NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MSC INTERNAL NOTE NO. 69-FM-36

February 7, 1969

RTCC REQUIREMENTS FOR MISSION G: MSFN TRACKING DATA PROCESSOR FOR POWERED FLIGHT LUNAR ASCENT/DESCENT NAVIGATION

TRW Systems Group

and

Mathematical Physics Branch

MISSION PLANNING AND ANALYSIS DIVISION

MANNED SPACECRAFT CENTER

HOUSTON, TEXAS

MSC INTERNAL NOTE NO. 69-FM-36

PROJECT APOLLO

RTCC REQUIREMENTS FOR MISSION G: MSFN TRACKING DATA PROCESSOR FOR POWERED FLIGHT LUNAR ASCENT/DESCENT NAVIGATION

By W. M. Lear, TRW Systems Group, and
Howard G. deVezin, Jr., Alan D. Wylie, and
Emil R. Schiesser, Mathematical Physics Branch

February 7, 1969

MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Approved: ______________________
James C. McPherson, Chief
Mathematical Physics Branch

Approved: ______________________
John P. Mayer, Chief
Mission Planning and Analysis Division

CONTENTS

Section Page

RTCC REQUIREMENTS FOR MISSION G: MSFN TRACKING DATA

PROCESSOR FOR POWERED FLIGHT LUNAR ASCENT/DESCENT NAVIGATION

By W. M. Lear, TRW Systems Group, and
Howard G. deVezin, Jr., Alan D. Wylie, and Emil R. Schiesser
Mathematical Physics Branch

1.0 INTRODUCTION

This report presents the basic RTCC requirements for a MSFN tracking data processor which is designed to process high-speed S-band tracking data to determine the position and velocity of the LM during powered descent and ascent from the lunar surface. These MSFN state vectors are used to determine whether the primary guidance and navigation control system (PGNCS) or the abort guidance system (AGS) is the more correct LM navigation system if the two significantly disagree.

The program will be used to process high-speed free-flight data as a special case of powered flight, but only for short arcs prior to descent and following ascent. The program is designed to process three-way and/or two-way Doppler tracking data simultaneously from one to four tracking stations. The expected tracking data rate is 10 measurements per second. The expected data processing rate is one measurement every 0.2 or 0.4 second. A processing rate of less than one measurement every 0.4 second (0.5, 0.6, etc.) might require revisions to the logic incorporated in this document.

2.0 PREPROCESSORS

2.1 Preorbit Determination MSFN High-Speed Data Processor

As the high-speed tracking data comes in, the pre-OD MSFN processor stores data into a table which is accessable to the powered-flight navigation (PFN) program. Data from one to four trackers is stored. The preprocessor performs gross-data editing (format checks), which can be the same as that described for the residual display processor in reference 1, if the "fine edit" is deleted. This edit routine is replaced by a new edit scheme whose logic is internal to the filter. The new edit routine is necessarily more restrictive in the labeling of data as valid when it is stored in the data tables. The tables should hold about 20 seconds of data for each tracker.

2.2 Telemetry Preprocessor

The PFN program will have a restart capability which depends on the use of LM-based position and velocity vectors received by telemetry. The preprocessor for storing telemetry vectors is described in reference 1.

3.0 BASIC METHOD AND EQUATIONS

The basic procedure is to process one to four Doppler observations (nominally four) at a given time to obtain a correction to the estimated position and velocity. Kalman filter equations are used to compute the correction. After the update, the state and its weight are propagated to the next set of observations to obtain another update.

The following is a general description of the basic Kalman filter equations used in the processor.

Let the subscript i denote time t_i, and let the subscript i/j denote an estimate at time t_i based on measurements made up to and including time t_j. Define $\underline{x}$ to be the state vector. The dynamic equations of motion are assumed to be in the form

$$\underline{x}_{i+1} = \underline{f}(\underline{x}_i, t_i, t_{i+1}) + \underline{r}_i$$

where $\underline{r}_i$ is the state noise vector satisfying

$$E[\underline{r}_i] = 0$$

$$E[\underline{r}_i \, \underline{r}_j^T] = 0 \qquad i \neq j$$

$E[\underline{r}_i \, \underline{r}_i^T] = R$ where R is the state noise covariance matrix.

Measurements or observations are assumed to be of the form

$$\underline{y}_i^* = \underline{g}(\underline{x}_i, t_i) + \underline{w}_i$$

where $\underline{w}_i$ is the measurement noise vector satisfying

$$E[\underline{w}_i] = 0$$

$$E[\underline{w}_i\ \underline{w}_j^T] = 0 \qquad i \neq j$$

$E[\underline{w}_i\ \underline{w}_i^T] = W$ where w is the measurement noise covariance matrix.

Define the state error covariance matrix by

$$J_{i/j} = E[(\underline{x}_i - \underline{x}_{i/j})(\underline{x}_i - \underline{x}_{i/j})^T]$$

Define the measurement matrix by

$$M = \left[\frac{\partial \underline{g}}{\partial \underline{x}_i}\right] \text{ evaluated at } \underline{x}_{i/i-1} \text{ and } t_i$$

Define the updating or transition matrix by

$$U = \left[\frac{\partial \underline{f}}{\partial \underline{x}_i}\right] \text{ evaluated at } \underline{x}_{i-1/i-1},\ t_i,\ t_{i-1}$$

Then the Kalman filter equations can be written as

$$\underline{x}_{i/i-1} = \underline{f}(\underline{x}_{i-1/i-1},\ t_i,\ t_{i-1}) \qquad (1a)$$

$$J_{i/i-1} = U\ J_{i-1/i-1}\ U^T + R \qquad (1b)$$

$$\underline{y}_{i/i-1} = \underline{g}(\underline{x}_{i/i-1},\ t_i) \qquad (1c)$$

$$B = J_{i/i-1} M^T (M J_{i/i-1} M^T + W)^{-1} \quad (1d)$$

$$\underline{x}_{i/i} = \underline{x}_{i/i-1} + B(\underline{y}_i^* - \underline{y}_{i/i-1}) \quad (1e)$$

$$J_{i/i} = (I - BM) J_{i/i-1} \quad (1f)$$

Initial values of $\underline{x}_{i-1/i-1}$ and $J_{i-1/i-1}$ must be specified to start the filter.

If programed directly, equations (1) would be inefficient in the use of machine time and core storage. The modified form of these equations [equations (2)] was used instead to generate the detailed programing instructions (section 14).

$$U = \partial \underline{f} / \partial \underline{x} \quad (2a)$$

$$\underline{x} = \underline{f}(\underline{x}, t_{i-1}, t_i) \quad (2b)$$

$$A = J U^T \quad (2c)$$

$$J = U A + R \quad (2d)$$

$$\hat{\underline{y}} = \underline{g}(\underline{x}, t_i) \quad (2e)$$

$$\hat{\underline{y}} = \underline{y}^* - \hat{\underline{y}} \quad (2f)$$

$$M = \partial \underline{g} / \partial \underline{x} \quad (2g)$$

$$D = J M^T \quad (2h)$$

$$H = MD + W \quad (2i)$$

$$H = H^{-1} \quad (2j)$$

$$B = DH \quad (2k)$$

$$\underline{x} = \underline{x} + B\hat{\underline{y}} \quad (2l)$$

$$J = J - B D^T \quad (2m)$$

4.0 THE STATE VECTOR DYNAMICS

The state vector $\underline{x}$, designated X(I) in the program, contains 21 elements.

$\underline{x}$ =

x	1	$\underline{R}_{V/E}$, **position of vehicle with respect to the earth**
y	2	
z	3	
$\dot{x}$	4	$\dot{\underline{R}}_{V/E}$
$\dot{y}$	5	
$\dot{z}$	6	
ψ_P	7	Pitch and yaw angles used to specify thrust direction
ψ_Y	8	
M	9	**Mass**
$\epsilon_{\omega P}$	10	Random variables adding to the nominal values of $\dot{\psi}_P$, $\dot{\psi}_Y$, $\dot{M}$, and specific impulse
$\epsilon_{\omega Y}$	11	
$\epsilon_{\dot{M}}$	12	
ϵ_I	13	
$\epsilon_{\omega 3,\ 1}$	14	**Random variables adding to the nominal values of the biasing** frequency, ω_3, at the receiving stations; i.e., rate bias errors
$\epsilon_{\omega 3,\ 2}$	15	
$\epsilon_{\omega 3,\ 3}$	16	
$\epsilon_{\omega 3,\ 4}$	17	
I_1	18	Constants of integration based on measurements in the form of integrated Doppler frequencies
I_2	19	
I_3	20	
I_4	21	

Let

$$\underline{R}_{M/E} = \begin{bmatrix} x_{M/E} \\ y_{M/E} \\ z_{M/E} \end{bmatrix}$$ Position of the moon with respect to the earth

$$\underline{C} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}$$ Unit vector in the direction of the thrust vector

Let I_N and $\dot{M}_N$ be the nominal values of specific impulse and mass flow rate. ($\dot{M}_N$ is a negative number.) Let $g = 19.92644969203518$ e.r./hr^2 (system parameter). Then the acceleration vector of a thrusting body in the earth-moon gravitational field is given by

$$\ddot{x} = -\frac{\mu_M(x - x_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_E\, x}{|\underline{R}_{V/E}|^3} - \frac{\mu_M\, x_{M/E}}{|\underline{R}_{M/E}|^3} - C_x \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

$$\ddot{y} = -\frac{\mu_M(y - y_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_E\, y}{|\underline{R}_{V/E}|^3} - \frac{\mu_M\, y_{M/E}}{|\underline{R}_{M/E}|^3} - C_y \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

$$\ddot{z} = -\frac{\mu_M(z - z_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_E\, z}{|\underline{R}_{V/E}|^3} - \frac{\mu_M\, z_{M/E}}{|\underline{R}_{M/E}|^3} - C_z \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

Higher-order gravitational harmonics are neglected since thrust model uncertainties are many times larger than the effect of these harmonics.

The unit vector $\underline{C}$ is obtained from the pitch and yaw angles in the following manner. The angles ψ_P and ψ_Y will be referenced to an arbitrary, nonrotating, thrust coordinate system as shown in figure 1. The u, v, w coordinates are referenced to x, y, z coordinates by means of the constant-coefficient, coordinate-transformation matrix shown below.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} C_1 & C_4 & C_7 \\ C_2 & C_5 & C_8 \\ C_3 & C_6 & C_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The $\underline{C}$ vector is now given by

$$\underline{C} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} C_1 & C_4 & C_7 \\ C_2 & C_5 & C_8 \\ C_3 & C_6 & C_9 \end{bmatrix} \begin{bmatrix} \cos \psi_Y \sin \psi_P \\ \cos \psi_Y \cos \psi_P \\ \sin \psi_Y \end{bmatrix}$$

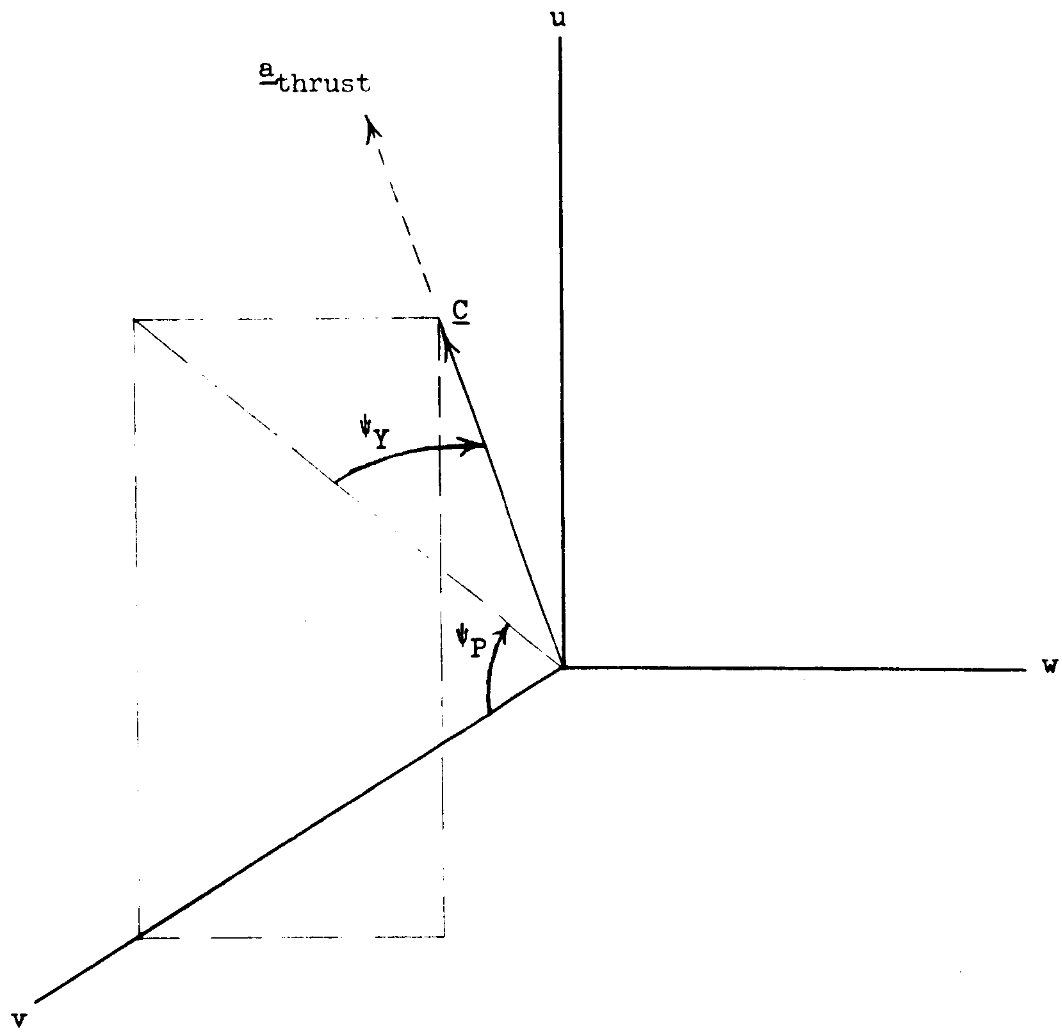


Figure 1.- Thrust coordinate system.

The coordinate transformation matrix will be chosen so as to make the angle ψ_Y small during descent and ascent. If ψ_Y gets large (close to ±90°), then $\dot{\psi}_P$ (or $\varepsilon_{\omega P}$) will also become large and ill-defined. This is undesirable because it is preferred to define the statistics of the random variable, $\varepsilon_{\omega P}$, as stationary in time; that is, independent of the geometry. Namely, since the maximum LM turning rate is limited by hardware to 15 deg/sec, the value of $\sigma_{\varepsilon\omega P}$ should be specified also to be approximately 15 deg/sec.

Choose the coordinate transformation matrix such that the initial thrust direction is easily expressed. This simplifies the choice of initial values for ψ_P and ψ_Y to start the filter. For the LM descent, the initial thrust direction is approximately in the direction of the negative velocity vector of the LM with respect to the moon. Also, the descent trajectory stays very close to the initial orbit plane of the LM. Thus, a logical choice for the u, v, w directions would be to choose the u direction to lie along the radius vector from the center of the moon to the LM near the start of descent. The v direction , or axis, would then be placed in the initial LM orbit plane and would point in the negative down-range direction; w would complete the right-hand triad. The initial values of ψ_Y and ψ_P both would be about zero, and ψ_Y would remain small during most of the descent.

For the LM ascent, the u axis will again be defined to be along the initial position vector with respect to the moon. However, this time, the v axis will be parallel to the CSM orbit plane (as opposed to the LM orbit plane for descent); w will complete the right-hand coordinate system. The initial value of ψ_P would be 90°, and the initial value of $\psi_Y = 0°$.

The mathematical equations for the unit vectors along the u, v, w directions are

$$\underline{i}_u = \frac{\underline{r}_1}{|\underline{r}_1|} \qquad \underline{i}_v = \frac{(\underline{r}_2 \times \underline{r}_3) \times \underline{i}_u}{|(\underline{r}_2 \times \underline{r}_3) \times \underline{i}_u|} \qquad \underline{i}_w = \underline{i}_u \times \underline{i}_v$$

Let the subscript V/M refer to vehicle (LM) with respect to the moon. Then

<u>for descent</u>

$$\underline{r}_1 = \underline{R}_{V/M}$$

$$\underline{r}_2 = \underline{R}_{V/M}$$

$$\underline{r}_3 = -\dot{\underline{R}}_{V/M}$$

<u>for ascent</u>

$$\underline{r}_1 = \underline{R}_{V/M}$$

$$\underline{r}_2 = \underline{R}_{CSM/M} \quad \text{(position of CSM)}$$

$$\underline{r}_3 = \underline{\dot{R}}_{CSM/M} \quad \text{(velocity of CSM)}$$

Note that

$$\underline{i}_u = C_1\,\underline{i}_x + C_2\,\underline{i}_y + C_3\,\underline{i}_z$$

$$\underline{i}_v = C_4\,\underline{i}_x + C_5\,\underline{i}_y + C_6\,\underline{i}_z$$

$$\underline{i}_w = C_7\,\underline{i}_x + C_8\,\underline{i}_y + C_9\,\underline{i}_z$$

where C_1 is the x component of $\underline{i}_u = \underline{r}_1/|\underline{r}_1|$, C_2 is the y component of $\underline{i}_u = \underline{r}_1/|\underline{r}_1|$, etc. Also note that

$$\underline{i}_x = C_1\,\underline{i}_u + C_4\,\underline{i}_v + C_7\,\underline{i}_w$$

$$\underline{i}_y = C_2\,\underline{i}_u + C_5\,\underline{i}_v + C_8\,\underline{i}_w$$

$$\underline{i}_z = C_3\,\underline{i}_u + C_6\,\underline{i}_v + C_9\,\underline{i}_w$$

To continue with the dynamics of the rest of the state variables, the dynamics of ψ_P, ψ_Y, and M are given by

$$\left.\begin{aligned} \dot{\psi}_P &= \omega_{PN} + \epsilon_{\omega P} \\ \dot{\psi}_Y &= \omega_{YN} + \epsilon_{\omega Y} \\ \dot{M} &= \dot{M}_N + \epsilon_{\dot{M}} \end{aligned}\right\} \quad \text{Assumed constant across the integration interval}$$

The subscript N stands for nominal value.

The dynamics of the random variables denoted by ϵ in the state vector (the 10th through the 17th element) are given by

$$\epsilon_{i+1} = \epsilon_i \exp(-|\Delta T|/T_\epsilon) + \sigma_\epsilon[1 -(\exp -|\Delta T|/T_\epsilon)^2]^{\frac{1}{2}} \eta_i$$

where

$$E[\eta_i] = 0$$

$$E[\eta_i \eta_j] = 0 \qquad i \neq j$$

$$E[\eta_i \eta_i] = 1$$

and where T_ϵ is the time constant associated with this random process. It can easily be shown that

$$E[\epsilon_j] = 0$$

$$E[\epsilon_j \epsilon_{j+k}] = \sigma_\epsilon^{\ 2} \exp(-|k \Delta T|/T_\epsilon)$$

That is, ϵ is an exponentially-correlated-in-time random variable; η is an uncorrelated-in-time forcing function for the difference equation for ϵ_{i+1}. Since the best estimate of the unknown function η is zero, ϵ is updated in the filter by

$$\epsilon_{i+1} = \epsilon_i \exp(-|\Delta T|/T_\epsilon)$$

The time constant, T_ϵ, gives a measure of how rapidly ϵ may be changing. T_ϵ equal to infinity implies that ϵ is a constant; T_ϵ equal to zero implies that ϵ is uncorrelated in time. Thus

$$\epsilon_{i+1} = \epsilon_i \times 1 = \epsilon_i \qquad \text{for } T_\epsilon \rightarrow \infty$$

$$\epsilon_{i+1} = \epsilon_i \times \text{zero} = 0 \qquad \text{for } T_\epsilon \rightarrow 0$$

$$\epsilon_{i+1} = \epsilon_i \times .905 \qquad \text{for } T_\epsilon = 10\ \Delta T$$

$$\epsilon_{i+1} = \epsilon_i \times .000045 \qquad \text{for } T_\epsilon = \Delta T/10$$

The last four elements of the state vector are assumed to be true constants; that is, $\dot{I}_L = 0$ for L = 1, 2, 3, 4.

5.0 INTEGRATING THE STATE VECTOR

Integration of $\underline{x}$ from t_i to t_{i+1} is accomplished by

$$x_{i+1} = x_i + \dot{x}_i \Delta T + \ddot{x}_i \Delta T^2/2 + \dddot{x}_i \Delta T^3/6$$

$$y_{i+1} = y_i + \dot{y}_i \Delta T + \ddot{y}_i \Delta T^2/2 + \dddot{y}_i \Delta T^3/6$$

$$z_{i+1} = z_i + \dot{z}_i \Delta T + \ddot{z}_i \Delta T^2/2 + \dddot{z}_i \Delta T^3/6$$

$$\dot{x}_{i+1} = \dot{x}_i + \ddot{x}_i \Delta T + \dddot{x}_i \Delta T^2/2$$

$$\dot{y}_{i+1} = \dot{y}_i + \ddot{y}_i \Delta T + \dddot{y}_i \Delta T^2/2$$

$$\dot{z}_{i+1} = \dot{z}_i + \ddot{z}_i \Delta T + \dddot{z}_i \Delta T^2/2$$

$$\psi_{P,\, i+1} = \psi_{P,\, i} + (\omega_{PN} + \epsilon_{\omega P,\, i})\, \Delta T^*$$

$$\psi_{Y,\, i+1} = \psi_{Y,\, i} + (\omega_{YN} + \epsilon_{\omega Y,\, i})\, \Delta T^*$$

$$M_{i+1} = M_i + (\dot{M}_N + \epsilon_{\dot{M},\, i})\, \Delta T^*$$

$$\epsilon_{\omega P,\, i+1} = \epsilon_{\omega P,\, i}[\exp(-|\Delta T|/T_{\omega P})]$$

$$\epsilon_{\omega Y,\, i+1} = \epsilon_{\omega Y,\, i}[\exp(-|\Delta T|/T_{\omega Y})]$$

$$\epsilon_{\dot{M},\, i+1} = \epsilon_{\dot{M},\, i}[\exp(-|\Delta T|/T_{\epsilon \dot{M}})]$$

$$\epsilon_{I,\ i+1} = \epsilon_{I,\ i}[\exp(-|\Delta T|/T_{\epsilon I})]$$

$$\epsilon_{\omega 31,\ i+1} = \epsilon_{\omega 31,\ i}[\exp(-|\Delta T|/T_{\omega 3})]$$

$$\epsilon_{\omega 32,\ i+1} = \epsilon_{\omega 32,\ i}[\exp(-|\Delta T|/T_{\omega 3})]$$

$$\epsilon_{\omega 33,\ i+1} = \epsilon_{\omega 33,\ i}[\exp(-|\Delta T|/T_{\omega 3})]$$

$$\epsilon_{\omega 34,\ i+1} = \epsilon_{\omega 34,\ i}[\exp(-|\Delta T|/T_{\omega 3})]$$

$$I_{1,\ i+1} = I_{1,\ i}$$

$$I_{2,\ i+1} = I_{2,\ i}$$

$$I_{3,\ i+1} = I_{3,\ i}$$

$$I_{4,\ i+1} = I_{4,\ i}$$

where ΔT^* is zero for free flight and $\Delta T^* = \Delta T$ for powered flight; ω_{PN} is the nominal value of $\dot{\psi}_P$, ω_{YN} is the nominal value of $\dot{\psi}_Y$, and $\dot{M}_N$ is the nominal value of $\dot{M}$.

The expressions for the second derivatives, $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, were given in the previous section. The third, time derivatives, are obtained from

$$\dddot{x} = \frac{\partial\ddot{x}}{\partial x}\dot{x} + \frac{\partial\ddot{x}}{\partial y}\dot{y} + \frac{\partial\ddot{x}}{\partial z}\dot{z} + \frac{\partial\ddot{x}}{\partial\psi_P}(\omega_{PN} + \epsilon_{\omega P}) + \frac{\partial\ddot{x}}{\partial\psi_Y}(\omega_{YN} + \epsilon_{\omega Y})$$

$$+ \frac{\partial\ddot{x}}{\partial M}(\dot{M}_N + \epsilon_{\dot{M}}) + \frac{\partial\ddot{x}}{\partial x_{M/E}}\dot{x}_{M/E} + \frac{\partial\ddot{x}}{\partial y_{M/E}}\dot{y}_{M/E} + \frac{\partial\ddot{x}}{\partial z_{M/E}}\dot{z}_{M/E}$$

$$\ddot{y} = \frac{\partial \ddot{y}}{\partial x}\dot{x} + \frac{\partial \ddot{y}}{\partial y}\dot{y} + \frac{\partial \ddot{y}}{\partial z}\dot{z} + \frac{\partial \ddot{y}}{\partial \psi_P}(\omega_{PN} + \epsilon_{\omega P}) + \frac{\partial \ddot{y}}{\partial \psi_Y}(\omega_{YN} + \epsilon_{\omega Y})$$

$$+ \frac{\partial \ddot{y}}{\partial M}(\dot{M}_N + \epsilon_{\dot{M}}) + \frac{\partial \ddot{y}}{\partial x_{M/E}}\dot{x}_{M/E} + \frac{\partial \ddot{y}}{\partial y_{M/E}}\dot{y}_{M/E} + \frac{\partial \ddot{y}}{\partial z_{M/E}}\dot{z}_{M/E}$$

$$\ddot{z} = \frac{\partial \ddot{z}}{\partial x}\dot{x} + \frac{\partial \ddot{z}}{\partial y}\dot{y} + \frac{\partial \ddot{z}}{\partial z}\dot{z} + \frac{\partial \ddot{z}}{\partial \psi_P}(\omega_{PN} + \epsilon_{\omega P}) + \frac{\partial \ddot{z}}{\partial \psi_Y}(\omega_{YN} + \epsilon_{\omega Y})$$

$$+ \frac{\partial \ddot{z}}{\partial M}(\dot{M}_N + \epsilon_{\dot{M}}) + \frac{\partial \ddot{z}}{\partial x_{M/E}}\dot{x}_{M/E} + \frac{\partial \ddot{z}}{\partial y_{M/E}}\dot{y}_{M/E} + \frac{\partial \ddot{z}}{\partial z_{M/E}}\dot{z}_{M/E}$$

The nonzero partial derivatives of $\ddot{x}$, $\ddot{y}$, $\ddot{z}$ with respect to the elements of the state vector are shown below.

$$\frac{\partial \ddot{x}}{\partial x} = -\frac{\mu_M}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_E}{|\underline{R}_{V/E}|^3} + 3\frac{\mu_M(x - x_{M/E})^2}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} + 3\frac{\mu_E\, x^2}{|\underline{R}_{V/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial y} = 3\frac{\mu_M(x - x_{M/E})(y - y_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} + 3\frac{\mu_E\, x\, y}{|\underline{R}_{V/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial z} = 3\frac{\mu_M(x - x_{M/E})(z - z_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} + 3\frac{\mu_E\, x\, z}{|\underline{R}_{V/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial \psi_P} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M}[-C_1 \cos\psi_Y \cos\psi_P + C_4 \cos\psi_Y \sin\psi_P]$$

$$\frac{\partial \ddot{x}}{\partial \psi_Y} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M}[C_1 \sin \psi_Y \sin \psi_P + C_4 \sin \psi_Y \cos \psi_P - C_7 \cos \psi_Y]$$

$$\frac{\partial \ddot{x}}{\partial M} = C_x \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} \frac{1}{M}$$

$$\frac{\partial \ddot{x}}{\partial x_{M/E}} = \frac{\mu_M}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_M}{|\underline{R}_{M/E}|^3} - 3 \frac{\mu_M (x - x_{M/E})^2}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} - 3 \frac{\mu_M x_{M/E}^2}{|\underline{R}_{M/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial y_{M/E}} = -3 \frac{\mu_M (x - x_{M/E})(y - y_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} - 3 \frac{\mu_M x_{M/E} y_{M/E}}{|\underline{R}_{M/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial z_{M/E}} = -3 \frac{\mu_M (x - x_{M/E})(z - z_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} - 3 \frac{\mu_M x_{M/E} z_{M/E}}{|\underline{R}_{M/E}|^5}$$

$$\frac{\partial \ddot{x}}{\partial \epsilon_{\dot{M}}} = -C_x \frac{g(I_N + \epsilon_I)}{M} \qquad \frac{\partial \ddot{x}}{\partial \epsilon_I} = -C_x \frac{g(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

$$\frac{\partial \ddot{y}}{\partial x} = \frac{\partial \ddot{x}}{\partial y}$$

$$\frac{\partial \ddot{y}}{\partial y} = -\frac{\mu_M}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_E}{|\underline{R}_{V/E}|^3} + 3 \frac{\mu_M (y - y_{M/E})^2}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} + 3 \frac{\mu_E y^2}{|\underline{R}_{V/E}|^5}$$

$$\frac{\partial \ddot{y}}{\partial z} = 3 \frac{\mu_M (y - y_{M/E})(z - z_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} + 3 \frac{\mu_E y z}{|\underline{R}_{V/E}|^5}$$

$$\frac{\partial \ddot{y}}{\partial \psi_P} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} [-C_2 \cos \psi_Y \cos \psi_P + C_5 \cos \psi_Y \sin \psi_P]$$

$$\frac{\partial \ddot{y}}{\partial \psi_Y} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} [C_2 \sin \psi_Y \sin \psi_P + C_5 \sin \psi_Y \cos \psi_P - C_8 \cos \psi_Y]$$

$$\frac{\partial \ddot{y}}{\partial M} = C_y \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} \frac{1}{M}$$

$$\frac{\partial \ddot{y}}{\partial x_{M/E}} = \frac{\partial \ddot{x}}{\partial y_{M/E}}$$

$$\frac{\partial \ddot{y}}{\partial y_{M/E}} = \frac{\mu_M}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^3} - \frac{\mu_M}{|\underline{R}_{M/E}|^3} - 3 \frac{\mu_M (y - y_{M/E})^2}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} - 3 \frac{\mu_M \, y_{M/E}^2}{|\underline{R}_{M/E}|^5}$$

$$\frac{\partial \ddot{y}}{\partial z_{M/E}} = -3 \frac{\mu_M (y - y_{M/E})(z - z_{M/E})}{|\underline{R}_{V/E} - \underline{R}_{M/E}|^5} - 3 \frac{\mu_M \, y_{M/E} \, z_{M/E}}{|\underline{R}_{M/E}|^5}$$

$$\frac{\partial \ddot{y}}{\partial \epsilon_{\dot{M}}} = -C_y \frac{g(I_N + \epsilon_I)}{M} \qquad \frac{\partial \ddot{y}}{\partial \epsilon_I} = -C_y \frac{g(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

$$\frac{\partial \ddot{z}}{\partial x} = \frac{\partial \ddot{x}}{\partial z} \qquad \frac{\partial \ddot{z}}{\partial y} = \frac{\partial \ddot{y}}{\partial z} \qquad \frac{\partial \ddot{z}}{\partial z} = - \frac{\partial \ddot{x}}{\partial x} - \frac{\partial \ddot{y}}{\partial y}$$

$$\frac{\partial \ddot{z}}{\partial \psi_P} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} [-C_3 \cos \psi_Y \cos \psi_P + C_6 \cos \psi_Y \sin \psi_P]$$

$$\frac{\partial \ddot{z}}{\partial \psi_Y} = \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} [C_3 \sin \psi_Y \sin \psi_P + C_6 \sin \psi_Y \cos \psi_P - C_9 \cos \psi_Y]$$

$$\frac{\partial \ddot{z}}{\partial M} = C_z \frac{g(I_N + \epsilon_I)(\dot{M}_N + \epsilon_{\dot{M}})}{M} \frac{1}{M}$$

$$\frac{\partial \ddot{z}}{\partial x_{M/E}} = \frac{\partial \ddot{x}}{\partial z_{M/E}} \qquad \frac{\partial \ddot{z}}{\partial y_{M/E}} = \frac{\partial \ddot{y}}{\partial z_{M/E}} \qquad \frac{\partial \ddot{z}}{\partial z_{M/E}} = -\frac{\partial \ddot{x}}{\partial x_{M/E}} - \frac{\partial \ddot{y}}{\partial y_{M/E}}$$

$$\frac{\partial \ddot{z}}{\partial \epsilon_{\dot{M}}} = -C_z \frac{g(I_N + \epsilon_I)}{M} \qquad \frac{\partial \ddot{z}}{\partial \epsilon_I} = -C_z \frac{g(\dot{M}_N + \epsilon_{\dot{M}})}{M}$$

In addition to integrating the state vector, the moon's position and velocity will also be integrated by

$$\underline{R}_{M/E,\ i+1} = \underline{R}_{M/E,\ i} + \dot{\underline{R}}_{M/E,\ i} \Delta T - \frac{\mu_E + \mu_M}{|\underline{R}_{M/E,\ i}|^3} \underline{R}_{M/E,\ i} \frac{\Delta T^2}{2}$$

$$\dot{\underline{R}}_{M/E,\ i+1} = \dot{\underline{R}}_{M/E,\ i} - \frac{\mu_E + \mu_M}{|\underline{R}_{M/E,\ i}|^3} \underline{R}_{M/E,\ i} \Delta T$$

6.0 PROPAGATING THE STATE COVARIANCE MATRIX (J)

The transition matrix is defined by

$$U = \frac{\partial \underline{X}_{i+1}}{\partial \underline{X}_i}$$

The equations for $\underline{X}_{i+1}$ as a function of $\underline{X}_i$ were given in the previous section. Thus U may be obtained by taking the partial derivatives of these functions. Neglecting terms higher than first order in ΔT, the U matrix may be conveniently partitioned into

$$U = \begin{bmatrix} U1_{13 \times 13} & 0 \\ 0 & U4_{8 \times 8} \end{bmatrix}$$

In more detail, U1 looks like

$$
U1 = \begin{bmatrix}
1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
m_{11} & m_{12} & m_{13} & 1 & 0 & 0 & m_{17} & m_{18} & m_{19} & 0 & 0 & m_{1,12} & m_{1,13} \\
m_{21} & m_{22} & m_{23} & 0 & 1 & 0 & m_{27} & m_{28} & m_{29} & 0 & 0 & m_{2,12} & m_{2,13} \\
m_{31} & m_{32} & m_{33} & 0 & 0 & 1 & m_{37} & m_{38} & m_{39} & 0 & 0 & m_{3,12} & m_{3,13} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T^* & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T^* & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T^* & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{10} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{11} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{13}
\end{bmatrix}
$$

where

$$m_{1,\,i} = \frac{\partial \ddot{x}}{\partial x,\ y,\ z,\ \psi_P,\ \psi_Y,\ M,\ \epsilon_{\dot{M}},\ \epsilon_I} \Delta T \qquad i = 1, 2, 3, 7, 8, 9, 12, 13$$

$$m_{2,\,i} = \frac{\partial \ddot{y}}{\partial x,\ y,\ z,\ \psi_P,\ \psi_Y,\ M,\ \epsilon_{\dot{M}},\ \epsilon_I} \Delta T \qquad i = 1, 2, 3, 7, 8, 9, 12, 13$$

$$m_{3,\,i} = \frac{\partial \ddot{z}}{\partial x,\ y,\ z,\ \psi_P,\ \psi_Y,\ M,\ \epsilon_{\dot{M}},\ \epsilon_I} \Delta T \qquad i = 1, 2, 3, 7, 8, 9, 12, 13$$

$$P_{10} = \exp(-|\Delta T|/T_{\omega P})$$

$$P_{11} = \exp(-|\Delta T|/T_{\omega Y})$$

$$P_{12} = \exp(-|\Delta T|/T_{\epsilon \dot{M}})$$

$$P_{13} = \exp(-|\Delta T|/T_{\epsilon I})$$

The U4 matrix looks like

$$U4 = \begin{bmatrix} P_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{14} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{14} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$P_{14} = \exp(-|\Delta T|/T_{\omega 3})$$

The matrix multiplication $J = UJ\,U^T$ is the most time consuming operation in the filter. Straightforward matrix multiplication in the above equation would require about 18 500 scalar multiplications and 18 500 scalar additions. Hence, the cycle time of the filter is greatly dependent on how efficiently $UJ\,U^T$ is evaluated.

As a first step toward efficient computations, the simplest of all possible U matrices was evaluated from first-order Taylor series expansions. This causes U to contain large blocks of zeros which will now be used to our advantage.

As presented on the previous page, U can be conveniently partitioned into four parts. Likewise, the 21 by 21 J matrix can conceptually (but not actually) be partitioned as

$$J = \begin{bmatrix} J1_{13 \times 13} & J2_{13 \times 8} \\ J3_{8 \times 13} & J4_{8 \times 8} \end{bmatrix}$$

Note that $J = J^T$, so $J2 = J3^T$. Thus, wherever J2 appears, $J3^T$ will be used in place of it.

$UJ\ U^T$ is given by

$$UJ\ U^T = \begin{bmatrix} U1 & 0 \\ 0 & U4 \end{bmatrix} \begin{bmatrix} J1 & J3^T \\ J3 & J4 \end{bmatrix} \begin{bmatrix} U1^T & 0 \\ 0 & U4^T \end{bmatrix} = \begin{bmatrix} U1\ J1\ U1^T & (U4\ J3\ U1^T)^T \\ U4\ J3\ U1^T & U4\ J4\ U4^T \end{bmatrix}$$

The 13 by 13 temporary storage matrix, A, is used to assist in the above matrix multiplications. Thus, $J1 = U1\ J1\ U1^T$ is given by

$$A = J1\ U1^T$$

$$J1 = U1\ A$$

By the use of the special nature of the U1 matrix, A is given by

```
DØ c  I  =  1, 13

DØ a  J  =  1, 3

A(I, J)  =  J(I, J) + J(I, J + 3)*ΔT
```

```
a     A(I, J+3)  =  J(I,1)*M(J,1) + J(I,2)*M(J,2) + J(I,3)*M(J,3)

                    + J(I,7)*M(J,7) + J(I,8)*M(J,8) + J(I,9)*M(J,9)

                    + J(I,12)*M(J,12) + J(I,13)*M(J,13) + J(I,J+3)

      DØ b   J = 7, 9

b     A(I, J)  =  J(I, J) + J(I, J+3)*ΔT*

      DØ c   J = 10, 13

c     A(I, J)  =  J(I, J)*P(J)
```

Again by the use of the special form of U1, the symmetric matrix J1 is given by

```
      DØ b   I = 1, 3

      DØ a   J = I, 13

a     J(I, J)  =  A(I, J) + A(I + 3, J)*ΔT

      L  =  I + 3

      DØ b   J = L, 13

b     J(L, J)  =  M(I,1)*A(1,J) + M(I,2)*A(2,J) + M(I,3)*A(3,J)

                  + M(I,7)*A(7,J) + M(I,8)*A(8,J) + M(I,9)*A(9,J)

                  + M(I,12)*A(12,J) + M(I,13)*A(13,J) + A(L,J)

      DØ c   I = 7, 9

      DØ c   J = I, 13
```

```
c     J(I, J)  =  A(I, J) + A(I +3, J)*ΔT*
      DØ d   I = 10, 13
      DØ d   J = I, 13
d     J(I, J)  =  P(I)*A(I, J)
      DØ e   I = 2, 13
      L  =  I - 1
      DØ e   J = 1, L
e     J(I, J)  =  J(J, I)
```

If A is used to store U4 J3, then

```
      DØ a   J = 1, 13
      DØ a   I  = 1, 4
      A(I, J)  =  P(14)*J(I + 13, J)
a     A(I + 4, J)  =  J(I + 17, J)
```

J3 = U4 J3 U1^T is now given by

```
      DØ c   I = 1, 8
      DO a   J = 1, 3
      J(I + 13, J)  =  A(I, J) + A(I, J+3)*ΔT
a     J(I+13, J+3)  =  A(I,1)*M(J,1) + A(I,2)*M(J,2) + A(I,3)*M(J,3)
                 + A(I,7)*M(J,7) + A(I,8)*M(J,8) + A(I,9)*M(J,9)
                 + A(I,12)*M(J,12) + A(I,13)*M(J,13) + A(I, J+3)
      DØ b   J = 7, 9
b     J(I+13, J)  =  A(I, J) + A(I, J+3)*ΔT*
      DØ c   J = 10, 13
c     J(I+13, J)  =  A(I, J)*P(J)
```

U4 J4 U4^T is given by

```
      E(1)  =  P(14)**2
      DØ a   I = 14, 17
      DØ a   J = I, 17
      J(I, J)  =  E(1)*J(I, J)
a     J(J, I)  =  J(I, J)
      DØ b   I = 18, 21
      DØ b   J = 14, 17
      J(I, J)  =  P(14)*J(I, J)
b     J(J, I)  =  J(I, J)
```

The use of the previous algorithms to compute $UJ U^T$ requires about 1100 floating point multiplications and about 900 floating point additions. These algorithms are also presented as part of the detailed programing instructions.

7.0 THE STATE NOISE (PREDICTOR MODEL UNCERTAINTIES)

To update the state vector from time t_i to t_{i+1}, symbolically use the equation

$$\underline{x}_{i+1} = \underline{f}(\underline{x}_i, t_i, t_{i+1})$$

However, since the true form of $\underline{f}$ is not known, use instead

$$\hat{\underline{x}}_{i+1} = \hat{\underline{f}}(\underline{x}_i, t_i, t_{i+1})$$

where the ^ stands for estimated value or function. Thus

$$\hat{\underline{x}}_{i+1} = \underline{x}_i + \underline{r}_i$$

where $\underline{r}_i = \hat{\underline{f}} - \underline{f}$ represents the model uncertainties or state noise.

Let a_{gx}, a_{gy}, a_{gz} be the unmodelled gravitational acceleration components. Then the first six elements of the $\underline{r}$ vector are

$$r_1 = a_{gx}\,\Delta T^2/2 \qquad r_2 = a_{gy}\,\Delta T^2/2 \qquad r_3 = a_{gz}\,\Delta T^2/2$$

$$r_4 = a_{gx}\,\Delta T \qquad r_5 = a_{gy}\,\Delta T \qquad r_6 = a_{gz}\,\Delta T$$

Since ΔT is on the order of 0.2 to 0.4 second, position error components, r_1, r_2, r_3 will be ignored. Note that velocity errors couple into position errors anyway during the next cycle in the filter, and, for a large number of integration steps, the final results are the same.

The state noise added to the exponentially correlated random variables is

$$r_{10} = \sigma_{\omega P}[1 - (\exp -|\Delta T|/T_{\omega P})^2]^{1/2}\ \eta_{\omega P}$$

$$r_{11} = \sigma_{\omega Y}[1 - (\exp -|\Delta T|/T_{\omega Y})^2]^{1/2} \quad \eta_{\omega Y}$$

$$r_{12} = \sigma_{\epsilon \dot{M}}[1 - (\exp -|\Delta T|/T_{\epsilon \dot{M}})^2]^{1/2} \quad \eta_{\epsilon \dot{M}}$$

$$r_{13} = \sigma_{\epsilon I}[1 - (\exp - |\Delta T|/T_{\epsilon I})^2]^{1/2} \quad \eta_{\epsilon I}$$

$$r_{14} = \sigma_{\omega 3}[1 - (\exp -|\Delta T|/T_{\omega 3})^2]^{1/2} \quad \eta_{\omega 3,1}$$

$$r_{15} = \sigma_{\omega 3}[1 - (\exp -|\Delta T|/T_{\omega 3})^2]^{1/2} \quad \eta_{\omega 3,2}$$

$$r_{16} = \sigma_{\omega 3}[1 - (\exp -|\Delta T|/T_{\omega 3})^2]^{1/2} \quad \eta_{\omega 3,3}$$

$$r_{17} = \sigma_{\omega 3}[1 - (\exp -|\Delta T|/T_{\omega 3})^2]^{1/2} \quad \eta_{\omega 3,4}$$

where the η's are zero-mean, unit-variance, uncorrelated (in time and with each other) random noise. All other components of $\underline{r}$ are zero.

The state noise covariance matrix is, by definition,

$$R = E[\underline{r}\ \underline{r}^T]$$

The nonzero elements of R are listed below.

$$R(4,4) = R(5,5) = R(6,6) = \sigma_{ag}^2 \Delta T^2$$

$$R(10,10) = \sigma_{\omega P}^2[1 -(\exp -|\Delta T|/T_{\omega P})^2]$$

$$R(11,11) = \sigma_{\omega Y}^2[1 -(\exp -|\Delta T|/T_{\omega Y})^2]$$

$$R(12,12) = \sigma_{\epsilon \dot{M}}^2[1 -(\exp -|\Delta T|/T_{\epsilon \dot{M}})^2]$$

$$R(13,13) = \sigma_{\epsilon I}^2[1 -(\exp -|\Delta T|/T_{\epsilon I})^2]$$

$$R(14,14) = R(15,15) = R(16,16) = R(17,17) = \sigma_{\omega 3}^2[1 -(\exp -|\Delta T|/T_{\omega 3})^2]$$

Since ΔT will be approximately constant, the above quantities may be input constants to the program.

8.0 THE MEASUREMENTS AND THEIR PARTIAL DERIVATIVES

The actual count from station L is modelled by

$$N_{FL}^{*} = \frac{\omega_4\, \nu_{tr}}{c}\left[\left|\underline{R}_V(T_{VL}) - \underline{R}_T(T_{TL})\right| + \left|\underline{R}_V(T_{VL}) - \underline{R}_{SL}(T_0)\right|\right]$$

$$+ (\omega_3 + b_L + \epsilon_{\omega 3L})(T_0 - T_{OIL}) - I_L + w_L$$

where $\omega_4\, \nu_{tr}/c$ and ω_3 are input constants; b_L is a previously solved for "rate bias" correction; w_L is the uncorrelated random error adding to the measurements and is primarily due to quantization error from the cycle counter. Its standard deviation is about one-third cycle.

T_0, the observation time, is the time at which the cycle counter is read. T_0 is the same for all stations. T_{OIL} is the station L observation time at which the first good cycle count is available to the filter. T_{VL} is the time at which the signal had to leave the vehicle in order to arrive at station L at time T_0. T_{TL} is the time at which the signal had to leave the transmitter in order to arrive at the vehicle at time T_{VL}.

$\underline{R}_V$ is the position vector of the vehicle with respect to the earth. $\underline{R}_T$ is the position vector of the transmitter. $\underline{R}_{SL}$ is the position vector of the receiving station L. I_L is the station L constant of integration.

$$I_L = \frac{\omega_4\, \nu_{tr}}{c}\left[\left|\underline{R}_V(T_{VL}) - \underline{R}_T(T_{TL})\right| + \left|\underline{R}_V(T_{VL}) - \underline{R}_{SL}(T_{OIL})\right|\right] - N_{FL}^{*}(T_{OIL})$$

I_L and T_{OIL} are reset every time the measurement flag goes from bad to good.

The partial derivatives of N^*_{FL} with respect to the elements of the state vector at time T_F are[a]

$$\frac{\partial N^*_{FL}}{\partial x,y,z(T_F)} = \frac{\omega_4\ \nu_{tr}}{c}\left[\frac{\underline{R}_V(T_{VL})-\underline{R}_T(T_{TL})}{|\underline{R}_V(T_{VL})-\underline{R}_T(T_{TL})|} + \frac{\underline{R}_V(T_{VL})-\underline{R}_{SL}(T_0)}{|\underline{R}_V(T_{VL})-\underline{R}_{SL}(T_0)|}\right]_{x,y,z \text{ component}}$$

$$\frac{\partial N^*_{FL}}{\partial \dot{x}, \dot{y}, \dot{z}(T_F)} = (T_{VL} - T_F)\frac{\partial N^*_{FL}}{\partial x, y, z\ (T_F)}$$

$$\frac{\partial N^*_{FL}}{\partial \epsilon_{\omega 3L}} = T_0 - T_{OIL} \qquad \frac{\partial N^*_{FL}}{\partial I_L} = -1$$

The above partial derivatives are stored in the M matrix as follows:

$$M(L,\ 1) = \frac{\partial N^*_{FL}}{\partial x} \qquad M(L,\ 2) = \frac{\partial N^*_{FL}}{\partial y} \qquad M(L,\ 3) = \frac{\partial N^*_{FL}}{\partial z}$$

$$M(L,\ 4) = \frac{\partial N^*_{FL}}{\partial \dot{x}} \qquad M(L,\ 5) = \frac{\partial N^*_{FL}}{\partial \dot{y}} \qquad M(L,\ 6) = \frac{\partial N^*_{FL}}{\partial \dot{z}}$$

$$M(L,\ L+13) = \frac{\partial N^*_{FL}}{\partial \epsilon_{\omega 3L}} \qquad M(L,\ L+17) = \frac{\partial N^*_{FL}}{\partial I_L}$$

where L = 1, 2, 3, 4. The form of the M matrix (zero and nonzero elements) is shown on the next page.

---

[a] T_F is the filter time tag of the state vector. T_F will be within 0.02 second of all the T_{VL}.

69-FM-36

Addressees:
IBM/J. Bednarcyk (5)
H. Norman
R. Sogard
R. Williams (6)
FS5/J. Stokes (without enclosure)
L. Dungan
T. Conway
E. Shinpaugh
FC/C. Charlesworth
FM/J. Mayer
H. W. Tindall
C. R. Huss
D. H. Owen, Jr.
R. P. Parten
Branch Chiefs
FM6/R. Regelbrugge
FM5/R. Ernull
FM7/R. Nobles
D. Beggs

cc:
Bellcomm/V. Mummert
J. Cappellari
W. Niedfeldt
R. Sperry
IBM/Library
TRW/Library
TRW/B. J. Gordon (7)
TRW/D. P. Johnson
BM6/Robert L. Phelps (2)
CF/W. J. North
EG/D. C. Cheatham
EG/R. G. Chilton
EG/R. A. Gardiner
KA/R. F. Thompson
KM/W. B. Evans
PA/G. M. Low
PD/A. Cohen
PD/O. E. Maynard
PD7/R. V. Battey
PD8/J. P. Loftus, Jr.
PD12/R. J. Ward
FA/C. C. Kraft, Jr.
FA/S. A. Sjoberg
FA/R. G. Rose
FA/C. C. Critzos
FC/E. F. Kranz (5)
FC/P. Shaffer
FL/J. B. Hammack (2)
FM/Report Preparation (25)
FM12/R. J. Andrews
FM15/M. A. Goodwin
FM4/E. R. Schiesser (10)
Author (30)
MSFC/M. Hoarst
/J. Haussler
LRC/A. Mayo
W. Mayo
GSFC/P. Schmid
J. Barsky
K. Squires
B. Bryant
P. Brumberg
B. Kruger
W. Kahn
ARC/G. Smith
TRW(Houston)/G. Riddle
T. Kenney
M. Shapiro
R. Vander Stucken
TRW(Los Angeles)/J. Wing
W. Lear (3)
W. Schroeder
P. Joseph
FM2/V. Hanssen
E. Dupnick

FM4:ADWylie:fdb

UNITED STATES GOVERNMENT

Memorandum

NASA - Manned Spacecraft Center
Mission Planning & Analysis Division

TO : See List Below

DATE: FEB 7 1969

69-FM47-27

FROM : FM/Mission Planning and Analysis Division

SUBJECT: RTCC Requirements for Mission G: MSFN Tracking Data Processor for Powered Flight Lunar Ascent/Descent Navigation

1. The enclosed internal note, 69-FM-36, presents the RTCC requirements for a high-speed MSFN tracking data processor for powered flight lunar ascent/descent navigation for Mission G.

2. This processor is designed to process powered flight data but is capable of processing short arcs of adjacent free-flight data prior to powered descent and after powered ascent.

3. Certain input constants are currently being modified by TRW and will be specified once the optimization studies are completed.

James C. McPherson, Chief
Mathematical Physics Branch

The Flight Software Branch concurs with the above recommendation and requests IBM to proceed accordingly.

James C. Stokes, Jr., Chief
Flight Software Branch

APPROVED BY:

John P. Mayer
Chief, Mission Planning
and Analysis Division

Enclosure

Distribution: (See attached page)

Distribution:
FM/Chief
Deputy Chief
Assistant Chief
Technical Assistant
Assistant for Advance Planning
Chief, MPSO (2)
Branch Chiefs
FM13/G. Michos
K. Henley
FM15/Editing
FM2/J. Alphin
E. Dupnick
V. Hanssen
FM5/R. Ernull
FS/Chief, Flight Support Division
FS5/Chief, Flight Software Branch
E. Shinpaugh
R. Allen
J. Saenz
B. Brady
J. Williams
M. Conway
L. Dungan
R. Spaulding
T. Price
G. Sabionski
J. Garman
FC/Chief, Flight Control Division
J. Greene
S. Bales
W. Boone
W. Stoval
D. Reed
FC5/J. Bostick
P. Shaffer
FC54/J. Llewellyn
C. Deiterich
FC55/E. Pavalka
FC56/C. Parker
K. Russell
EG7/J. Hanaway
GSFC-Code 550/F. Vonbun
R. Groves
J. Siry
Code 830/J. J. Donegan
Code 832/J. Barsky (2)

TRW/Houston/D. G. Saile (3)
L. J. Skidmore
J. B. Clifford
R. J. Boudreau
J. R. Henson
A. L. Satin
A. C. Bond
J. L. Knoedler
O. R. Bergman
J. N. Dashiell
P. Norris
R. Vander Stucken
M. Shapiro
TRW/LA/B. Lear
G. Buis
P. Joseph
B. Osgood
IBM/G. Mueller
G. Warner
L. Murrell
A. Kircher
F. Cross
L. Morrison
J. Hoskins (5)

OPTIONAL FORM NO. 10
MAY 1962 EDITION
GSA FPMR (41 CFR) 101-11.6

UNITED STATES GOVERNMENT

Memorandum

NASA-Manned Spacecraft Center
Mission Planning & Analysis Division

TO : See List Below

DATE: SEP 11 1969

File No. 69-FM47-284

FROM : FM/Mission Planning and Analysis Division

SUBJECT: Formulation revisions for the RTCC powered flight MSFN navigation program for lunar descent/ascent

1. Reference: MSC Internal Note No. 69-FM-36, "RTCC Requirements for Mission G: MSFN Tracking Data Processor for Powered Flight Lunar Ascent/Descent Navigation," Lear, W. M.; deVezin, H. G.; Wylie, A. D.; Schiesser, E. R.; February 7, 1969.

2. The purpose of this memorandum is to propose a change to the formulation of the referenced RTCC requirements document. The change involves revising the criteria for editing the high speed MSFN Doppler data to be processed by the powered flight MSFN navigation program for lunar descent/ascent.

3. The necessity of the change was observed during Apollo 11. At certain intervals in both the descent and ascent phases, bad MSFN Doppler data passed the current edit criteria and caused the filter to become unstable, thus necessitating a program restart. This data has been passed through the filter with the proposed edit scheme, and no program restarts were necessary for either the descent or ascent phase.

James C. McPherson, Chief
Mathematical Physics Branch

The Flight Software Branch concurs with the above recommendation and requests IBM to proceed accordingly.

James C. Stokes, Jr., Chief
Flight Software Branch

APPROVED BY:

for John P. Mayer
Chief, Mission Planning
and Analysis Division

Attachment

Distribution: (See attached list)

FM4:ALWylie:rmr

CHANGE SHEET

FOR

MSC INTERNAL NOTE 69-FM-36 DATED FEBRUARY 7, 1969

RTCC REQUIREMENTS FOR MISSION G: MSFN TRACKING DATA PROCESSOR FOR POWERED FLIGHT LUNAR ASCENT/DESCENT NAVIGATION

By W. M. Lear, TRW Systems Group, and

Howard G. deVezin, Jr., Alan D. Wylie,

and Emil R. Schiesser, Mathematical

Physics Branch

Change 1

September 8, 1969

James C. McPherson, Chief
Mathematical Physics Branch

for John P. Mayer, Chief
Mission Planning and Analysis
Division

Page 1 of 7
(with enclosures)

Note: A black bar in the margin indicates the area of change.

After the attached enclosures, which are replacement pages, have been inserted, place this CHANGE SHEET between the cover and title page and write on the cover "CHANGE 1 inserted".

1. Replace pages 40, 55, 61, 91, and 92.

Change no.	Date	Description
1	9/8/69	The criteria for editing the high speed MSFN Doppler data are revised to require the program to test according to the magnitude of the MSFN residuals rather than according to the size of the MSFN rate biases.

$$\sigma_{\Delta 2} = \sqrt{6}\ \sigma_N \qquad (\sigma_N \approx \tfrac{1}{3} \text{ cycle})$$

$$\approx .82 \text{ cycles}$$

Because the random error is primarily caused by a zero to 1 quantization error in the cycle counter, the maximum value of $\Delta_2\ N_0$ caused by quantization error would be

$$|\Delta_2\ N_0| < 2 \text{ cycles}$$

Thus, the value for Δ_{2MAX} may be chosen as

$$\Delta_{2MAX} \approx 2 + 140\ \Delta T_0^{\ 2} \text{ cycles}$$

In summary the current data is accepted if

$$\Delta_{1MIN} < N_0 - N_1 < \Delta_{1MAX}$$

and

$$|(N_0 - N_1) - (N_1 - N_2)| < \Delta_{2MAX}$$

where

$$\Delta_{1MIN} \approx (10^6 - 28{,}000)\ \Delta T_0 \text{ cycles}$$

$$\Delta_{1MAX} \approx (10^6 + 28{,}000)\ \Delta T_0 \text{ cycles}$$

$$\Delta_{2MAX} \approx 2 + 140\ \Delta T_0^{\ 2} \text{ cycles}$$

There is one further data editing check performed in another part of the program. Immediately before the state vector is corrected, the solution for the residual is checked to see if it is too large, based on the current measurements. If it is too large, then the residual for that station is set to zero, the corresponding row of the measurement matrix is zeroed, F(L+8) is set to zero, and the measurement weighting matrix, B, is reestimated. The reason for this additional editing check is that with the normal editing a large rate bias error cannot be easily detected. Also, depending on the size of ΔT_0, a few wild data points can pass the normal editing.

The data status checks do the following

1. If data for station L is missing or bad for time T_0, the number in the station L cycle count cell is labeled bad by setting F(L+4) = 0.

2. Likewise, F(L+8) is set to zero (elsewhere in the program) if station L has too large a residual.

3. F(L) is set to 1 if station L data goes from bad to good. When F(L) = 1, T_{OIL} and the station L integration constant, I_L, will be reinitialized. Also the corresponding row and column of the state error covariance matrix will be reinitialized.

4. F(L) is set to -1 if the station L location changes or if the transmitter location changes. When F(L) = -1, the I_L, T_{OIL}, and $\varepsilon_{\omega 3L}$ = X(L+13) are reinitialized, and the corresponding row and column of the state error covariance matrix will be reinitialized.

5. The F(L+12) flag is set to zero or 1 according to whether the station is a 3-way or 2-way station.

A block diagram of this logic is shown in figure 3. In this diagram

YY(L) is storage for the current cycle count from station L

$YY_p(L)$ is the past value of YY(L)

$$\Delta YY(L) = YY(L) - YY_p(L)$$

$$\Delta YY_p(L) = \text{past value of } \Delta YY(L)$$

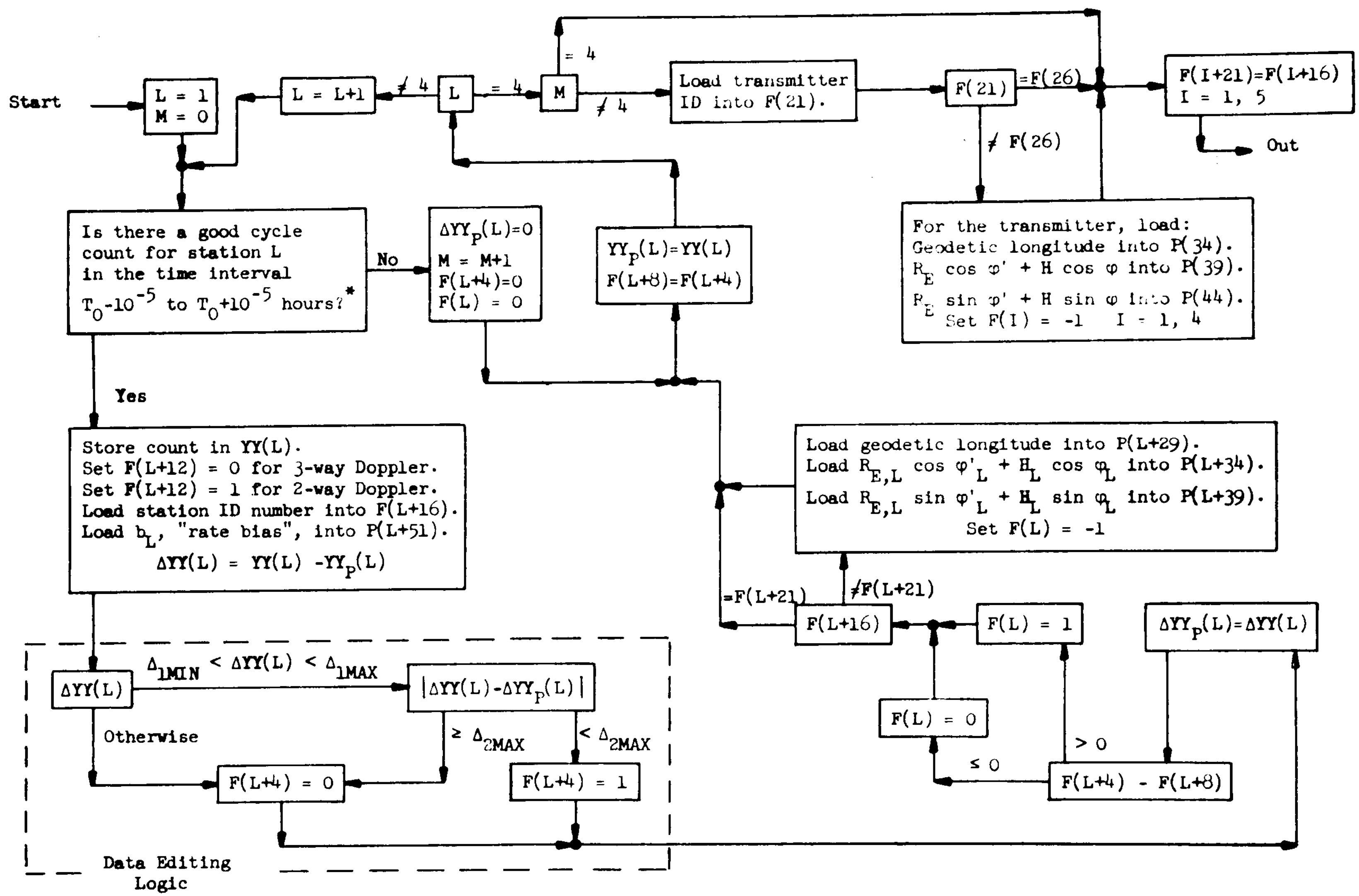


Figure 3.- Block diagram for data input logic and data status checks.

*The word good means that the tracking station attached a good label to the data.

10.0 INITIALIZATION OF MEASUREMENT CONSTANTS

When $F(L) = 1$, T_{OIL} and the station L integration constant, I_L, will be reinitialized. Also, the corresponding row and column of the state error covariance matrix will be reinitialized. In section 8 the initialization equation for I_L was given by

$$I_L = \frac{\omega_4 \nu_{tr}}{c} \left[\left| \underline{R}_V(T_{VL}) - \underline{R}_T(T_{TL}) \right| + \left| \underline{R}_V(T_{VL}) - \underline{R}_{SL}(T_{OIL}) \right| \right] - N^{*}_{FL}(T_{OIL})$$

When $F(L) = -1$, T_{OIL}, I_L and $\varepsilon_{\omega 3L} = X(L + 13)$ will be reinitialized; $X(L + 13)$ is initialized by setting it equal to zero. Likewise, the corresponding rows and columns of the state error covariance will be reinitialized. Whenever initialization takes place, the measurement is used in the initialization equations and should not be used again in the Kalman filter. To prevent this reuse, the appropriate row of the M matrix is zeroed out.

The block diagram for the initialization procedure is shown in figure 4. In this block diagram

$P(46) = \sigma^2_{\omega 3}$ for a 3-way station

$P(47) = \sigma^2_{\omega 3}$ for a 2-way station

$P(L+47) = \sigma^2_{\omega 3}$ for station L, L = 1, 2, 3, 4

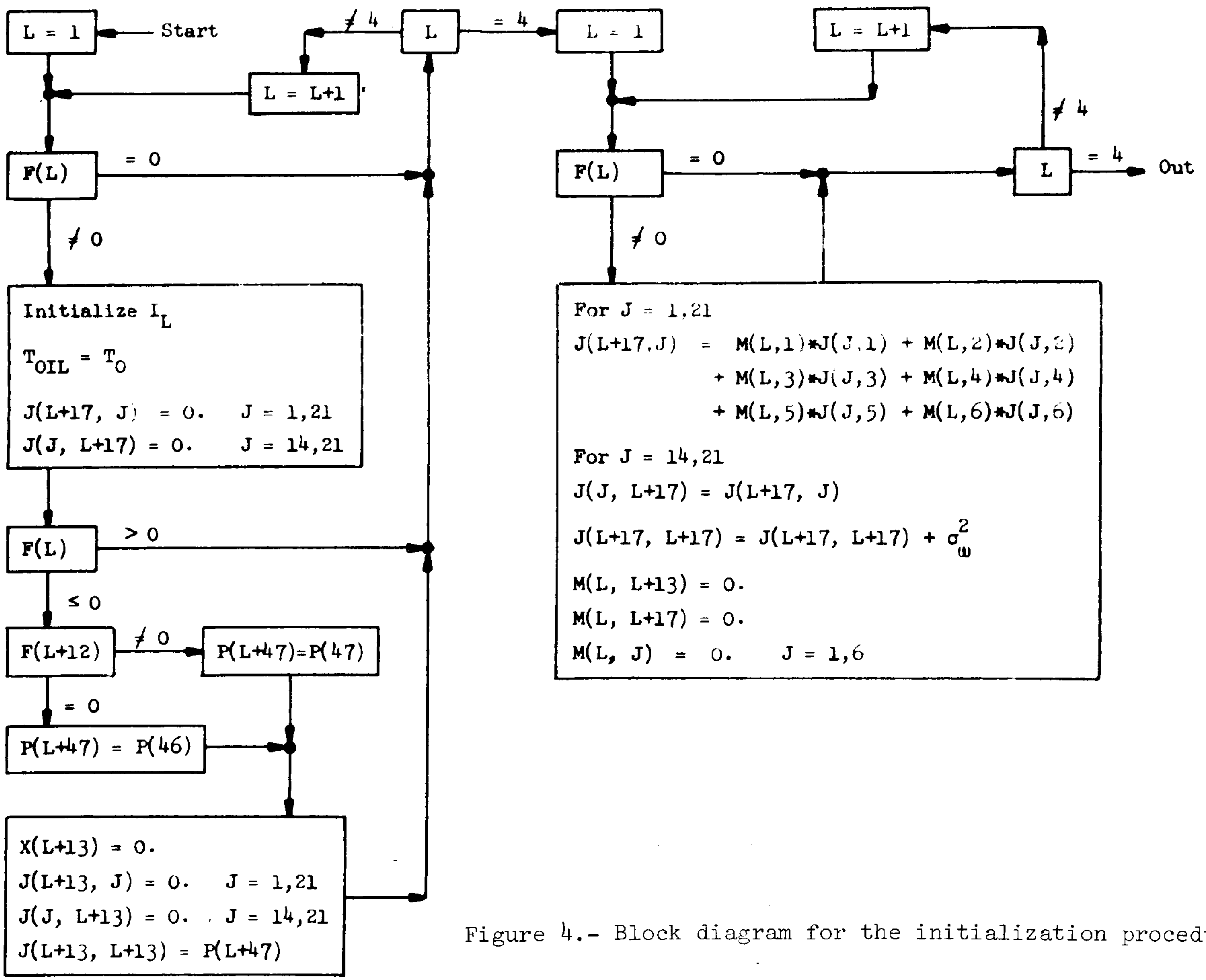


Figure 4.- Block diagram for the initialization procedure.

11.0 STATE AND COVARIANCE MATRIX UPDATE EQUATIONS

As in section 6, special procedures to save time are used to perform the following matrix operations: $D = JM^T$, $H = MD + W$, $H = H^{-1}$, $B = DH$, $J = J - BD^T$, and $\underline{X} = \underline{X} + B(\underline{Y}^* - \underline{\hat{Y}})$. The special procedures are given below and as part of the detailed program instructions.

In section 8, the M matrix was shown in the form

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{1,14} & 0 & 0 & 0 & m_{1,18} & 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{2,15} & 0 & 0 & 0 & m_{2,19} & 0 & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{3,16} & 0 & 0 & 0 & m_{3,20} & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{4,17} & 0 & 0 & 0 & m_{4,21} \end{bmatrix}$$

By the use of the special form of the M matrix, $D = J\,M^T$ is given by

```
DØ a   I = 1,21
DØ a   J = 1,4
```

```
a    D(I,J)  =  J(I,1)*M(J,1) + J(I,2)*M(J,2) + J(I,3)*M(J,3)
                + J(I,4)*M(J,4) + J(I,5)*M(J,5) + J(I,6)*M(J,6)
                + J(J+13, I)*M(J, J+13) + J(J+17, I)* M(J, J+17)
```

Note that J(I,J) I = 1,13 J = 14,21 is not used in the above algorithm.

H = MD + W is given by

```
     DØ b   I = 1,4
     DØ a   J = 1,I
a    H(I,J)  =  M(I,1)*D(1,J) + M(I,2)*D(2,J) + M(I,3)*D(3,J)
                + M(I,4)*D(4,J) + M(I,5)*D(5,J) + M(I,6)*D(6,J)
                + M(I, I+13)*D(I+13, J) + M(I, I+17)*D(I+17, J)
```

$$b \quad H(I,I) = H(I,I) + \sigma_w^2$$

Note that only the lower triangular part of the symmetric matrix H is calculated.

$H = H^{-1}$ is given by

```
     H(1,1)  =  1.DO/H(1,1)
     DØ e   J = 1,3
     L = J+1
     DØ a   I = 1,J
     H(I,L)  =  O. DO
     DØ a   K = 1,J
a    H(I,L)  =  H(I,L) - H(L,K)*H(I,K)
     DØ b   K = 1,J
```

```
b     H(L,L)  =  H(L,L) + H(L,K)*H(K,L)
      H(L,L)  =  1. DO/H(L,L)
      DØ c   I = 1,J
c     H(L,I)  =  H(I,L)*H(L,L)
      DØ d   I = 1,J
      DØ d   K = I,J
      H(I,K)  =  H(I,K) + H(I,L)*H(L,K)
d     H(K,I)  =  H(I,K)
      DØ e   I = 1,J
e     H(I,L)  =  H(L,I)
```

Note that the above inversion algorithm is designed specifically to invert a symmetric matrix using only the lower triangular part of that matrix. The algorithm requires about half the number of multiplications and additions that a normal matrix multiplication requires. Also, no external storage is used.

B = DH is given by

```
      DØ a   I = 1,21
      DØ a   J = 1,4
a     B(I,J)  =  D(I,1)*H(1,J) + D(I,2)*H(2,J) + D(I,3)*H(3,J)
                 + D(I,4)*H(4,J)
```

J = J - B D^T is given by

```
      DØ a   I = 1,13
      DØ a   J = I,13
```

```
    J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                 -B(I,4)*D(J,4)
a   J(J,I)  =  J(I,J)
    DØ c   I = 14,21
    DØ b   J = 1,13
b   J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                 -B(I,4)*D(J,4)
    DØ c   J = I,21
    J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                 -B(I,4)*D(J,4)
c   J(J,I)  =  J(I,J)
```

Note that the above algorithm makes use of the symmetry of the J matrix, and, again, J(I,J) I = 1, 13 J = 14, 21 is not used or disturbed. This part of the J matrix has not been used anywhere in the previous sections. Thus, this block of 104 double precision words of nondestroyable storage is available for other uses within the program.

$\underline{x} = \underline{x} + B(\underline{y}^* - \hat{\underline{y}})$, where $\underline{y}^* - \hat{\underline{y}}$ is stored in Y(I), is given by

```
DØ a   I = 1,21
X(I)  =  X(I) + B(I,1)*Y(1) + B(I,2)*Y(2) + B(I,3)*Y(3)
           + B(I,4)*Y(4)
```

12.0 LIGHT TIME SUBROUTINE (SUBROUTINE TRANST)

This subroutine solves the transit time equations; that is, given a measurement observation time T_0, TRANST solves for T_V, the time at which the signal had to leave the vehicle in order to arrive at the receiving station at time T_0. TRANST also solves for T_T, the time at which the signal had to leave the transmitter in order to arrive at the vehicle at time T_V.

The subroutine estimates $\underline{R}_V(T_{VI}) - \underline{R}_{SI}(T_0)$, the range vector from the receiver to the vehicle, and $\underline{R}_V(T_{VI}) - \underline{R}_T(T_{TI})$, the range vector from the transmitter to the vehicle. TRANST also solves for the magnitude of these vectors.

The time for station I, T_{VI}, is obtained by iterating the equation

$$T_{VI} = T_0 - \frac{1}{c}\Big\{[x(T_F) + (T_{VI} - T_F)\,\dot{x}(T_F) - x_{SI}(T_0)]^2 + [y(T_F) + (T_{VI} - T_F)\,\dot{y}(T_F) - y_{SI}(T_0)]^2 + [z(T_F) + (T_{VI} - T_F)\,\dot{z}(T_F) - z_{SI}(T_0)]^2\Big\}^{1/2}$$

A value of $T_{VI} = T_F$ (T_F is the filter time tag of the state vector) is used to start the iteration. Because of the manner in which T_F is updated ($\Delta T_F = T_{V1} + \Delta T_0 - T_F$), $T_{VI} - T_F$ is always less than 0.02 second.

From the above equation, $T_{VI}^{(n)}$ may be written symbolically for the nth iterated solution as

$$T_{VI}^{(n)} = f[T_{VI}^{(n-1)}]$$

The error, e, in the solution for T_{VI}, after n iterations, is given by the equation

$$|e^{(n)}| = |e^{(0)}| M^n$$

where M is the maximum absolute value of df/dT_{VI} in the vicinity of T_{VI}. The equation for df/dT_{VI} is

$$\frac{df}{dT_{VI}} = -\frac{1}{c}\frac{\underline{R}_{V/S} \cdot \dot{\underline{R}}_{V/E}}{|\underline{R}_{V/S}|}$$

where V/S refers to vehicle with respect to the station, and V/E refers to vehicle with respect to the earth. For a maximum vehicle velocity of 10^4 fps, M can be evaluated as

$$M \leq \frac{10^4}{10^9} = 10^{-5}$$

Because $|e^{(0)}|$ is less than $2 \cdot 10^{-2}$ seconds

$$|e^{(n)}| \leq 2 \cdot 10^{-2} (10^{-5})^n$$

for which a value of n = 1 will be used. Thus

$$|e^{(1)}| \leq 2 \cdot 10^{-7} \text{ seconds}$$

After T_{VI} is determined, the position of the vehicle can be calculated in the following manner.

$$x(T_{VI}) = x(T_F) + (T_{VI} - T_F)\,\dot{x}(T_F)$$

$$y(T_{VI}) = y(T_F) + (T_{VI} - T_F)\,\dot{y}(T_F)$$

$$z(T_{VI}) = z(T_F) + (T_{VI} - T_F)\,\dot{z}(T_F)$$

Also, the vector components $x(T_{VI}) - x_{SI}(T_0)$, $y(T_{VI}) - y_{SI}(T_0)$, $z(T_{VI}) - z_{SI}(T_0)$ and the absolute magnitude of this vector will be calculated.

In the calculation of T_{VI}, since the state vector is in MNBY coordinates, the four station location vectors must also be in MNBY coordinates. In true-of-date coordinates, the station location is given by

$$x'_{SI}(T_0) = P(I+34)*\cos[P(I+29) + \omega_{Earth} T_0]$$

$$y'_{SI}(T_0) = P(I+34)*\sin[P(I+29) + \omega_{Earth} T_0]$$

$$z'_{SI}(T_0) = P(I+39)$$

where P(I+29), P(I+34), and P(I+39) are obtained from the station characteristics table (section 9, fig. 2). These locations are then transformed to MNBY coordinates by the RNP matrix. This requires four coordinate transformations (one for each of the four stations).

$$\begin{bmatrix} x_{SI}(T_0) \\ y_{SI}(T_0) \\ z_{SI}(T_0) \end{bmatrix} = (RNP)^T \begin{bmatrix} x'_{SI}(T_0) \\ y'_{SI}(T_0) \\ z'_{SI}(T_0) \end{bmatrix}$$

The time T_{TI} is obtained from

$$T_{TI} = T_{VI} - \frac{1}{c}\left[[x(T_{VI}) - x_T(\hat{T}_T)]^2 + [y(T_{VI}) - y_T(\hat{T}_T)]^2 + [z(T_{VI}) - z_T(\hat{T}_T)]^2 \right]^{1/2}$$

where

$$\hat{T}_T = T_{V1} - (T_0 - T_{V1})$$

and where

$$\begin{bmatrix} x_T(\hat{T}_T) \\ y_T(\hat{T}_T) \\ z_T(\hat{T}_T) \end{bmatrix} = (RNP)^T \begin{bmatrix} P(39)*\cos[P(34) + \omega_E \hat{T}_T] \\ P(39)*\sin[P(34) + \omega_E \hat{T}_T] \\ P(44) \end{bmatrix}$$

Note that

$$\begin{bmatrix} \dot{x}_T(\hat{T}_T) \\ \dot{y}_T(\hat{T}_T) \\ \dot{z}_T(\hat{T}_T) \end{bmatrix} = (RNP)^T \begin{bmatrix} -\omega_E\ P(39)\ \sin[P(34) + \omega_E \hat{T}_T] \\ \omega_E\ P(39)\ \cos[P(34) + \omega_E \hat{T}_T] \\ 0 \end{bmatrix}$$

Thus, $x_T(T_{TI})$, $y_T(T_{TI})$, and $z_T(T_{TI})$ are given by

$$\begin{bmatrix} x_T(T_{TI}) \\ y_T(T_{TI}) \\ z_T(T_{TI}) \end{bmatrix} = \begin{bmatrix} x_T(\hat{T}_T) \\ y_T(\hat{T}_T) \\ z_T(\hat{T}_T) \end{bmatrix} + (T_{TI} - \hat{T}_T) \begin{bmatrix} \dot{x}_T(\hat{T}_T) \\ \dot{y}_T(\hat{T}_T) \\ \dot{z}_T(\hat{T}_T) \end{bmatrix}$$

The vector components $x(T_{VI}) - x_T(T_{TI})$, $y(T_{VI}) - y_T(T_{TI})$, $z(T_{VI}) - z_T(T_{TI})$ and the absolute magnitude of this vector can now be calculated.

This method for solving the transit time equations requires only six multiplications by the RNP matrix and only five calls to the sine and cosine subroutines. A more straightforward approach would have resulted in 12 RNP multiplications and 12 calls to the sine and cosine subroutines.

Despite the fact that only a single iteration is used to obtain T_{VI} and T_{TI}, the accuracy of the solutions is more than adequate. Based on a maximum vehicle velocity of 10^4 fps with respect to the earth, the maximum error in T_{VI} is $2 \cdot 10^{-7}$ seconds. The maximum error in calculating $\underline{R}_V(T_{VI}) - \underline{R}_{SI}(T_0)$ is 0.002 ft. The maximum error in calculating T_{TI} is about $0.4 \cdot 10^{-7}$ second.

13.0 SUPERVISORY LOGIC

The overall block diagram of the program is shown in figure 5. The program has four main areas of specialties. Blocks 1 and 2 are first-time-only blocks, used to initialize various constants and variables. Some of these quantities are reset in the restart block in the event that restart conditions occur. Blocks 3 through 6 are concerned with the dynamics. Blocks 7 through 14 process the measurements. Blocks 15 and 16 integrate the current best estimates of position and velocity ahead, or back, to a desired time, T_D. The output position and velocity in block 16 will be in MNBY selenocentric coordinates with units of earth radii and earth radii per hour. The time tag will be in hours from midnight of the launch day.

The subroutine TRAJ, mentioned in the overall block diagram, integrates the first six elements of the state vector over the interval ΔT seconds, updates time, and integrates the moon's position and velocity over the interval ΔT seconds. Subroutine TRAJ has its maximum integration step size limited to ΔT_{MAX}. This is to prevent an excessive truncation error in the state estimate if an unusually large ΔT is called for. A block diagram that illustrates how ΔT is limited is shown in figure 6.

New variables used in the block diagram are defined below.

T_F = filter time tag of the state vector

T_{RL} = real (current) ground time, hours from midnight of launch day

T_{V1} = time at which a signal would have to leave the spacecraft in order to arrive at the first ground tracking station at time T_0, the observation time

$(T_F+P_{119})R$ = T_F+P_{119} rounded to the nearest 2/36 000 hour (nearest 2/10 second) or nearest 4/36 000 hour, depending upon whether data is processed every 0.2 second or every 0.4 second

P_{119} = approximate value of $T_0 - T_{V1}$ (P_{119} is an input constant)

T_0 = observation time

T_L = constant lag time between observations and T_{RL}

ΔT_0 = time interval between observations

ΔT_F	=	integration step size for the filter
ΔT	=	integration step size used in TRAJ
ΔT_{MAX}	=	maximum allowable value of ΔT
T_D	=	desired time of output state vector
T_{DL}	=	last cycle's value of T_D

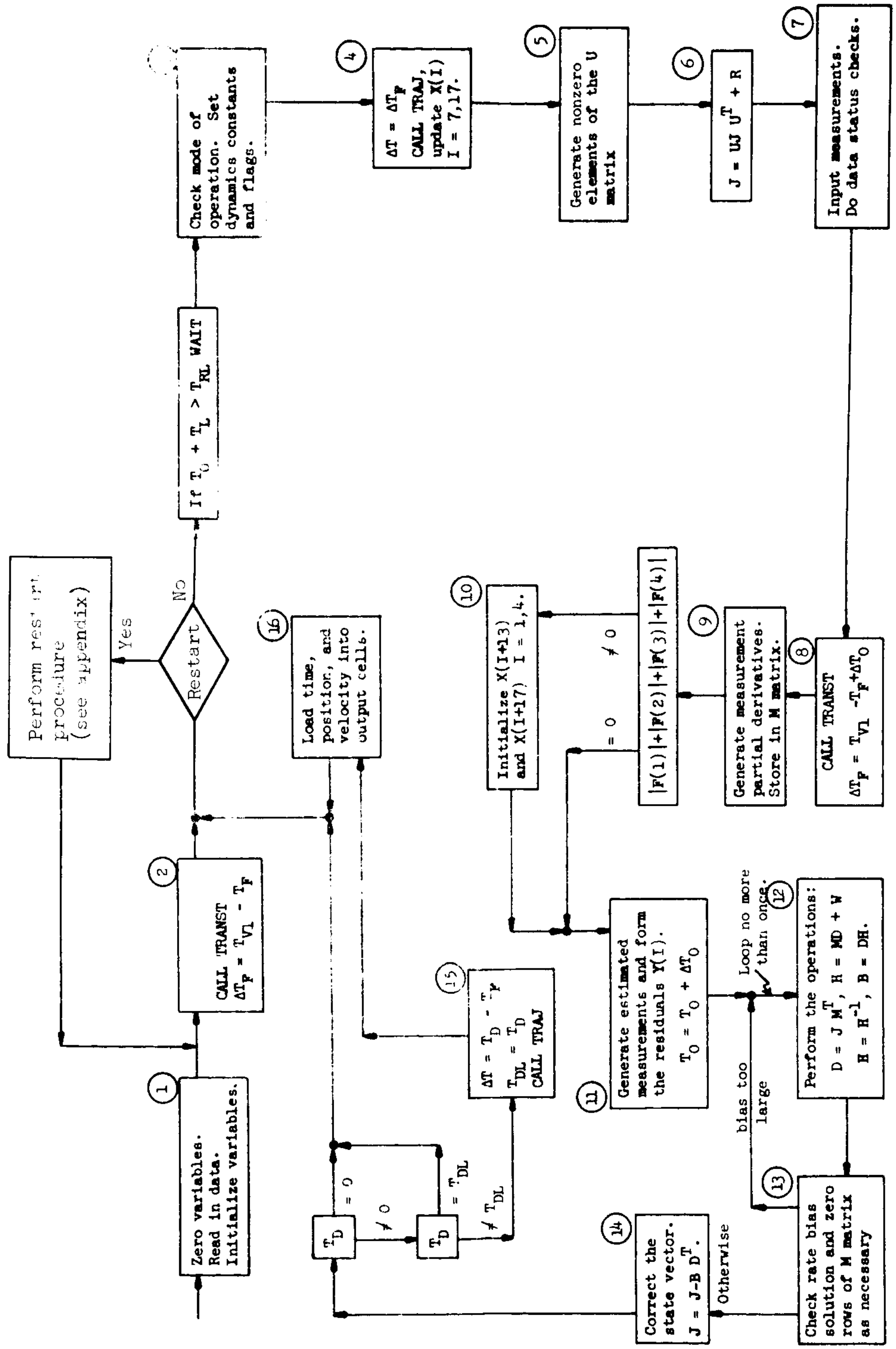


Figure 5.- Overall block diagram.

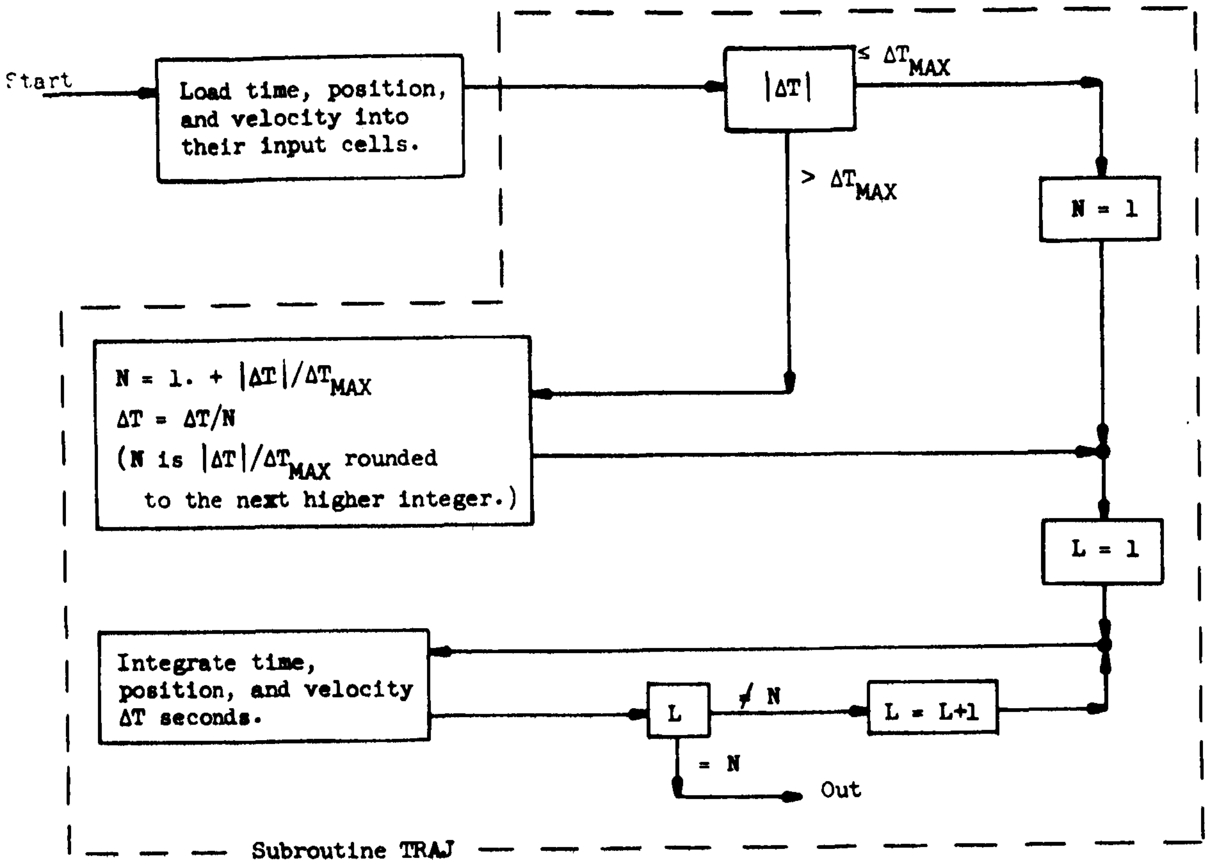


Figure 6.- Logic for controlling integration step size in subroutine TRAJ.

14.0 DETAILED PROGRAMING INSTRUCTIONS

The previous sections of this report have outlined the engineering equations and logic and were intended only to supply general background information. The following sections present the actual detailed programing requirements.

Ordinarily, a 21-state variable filter for use in real-time is indeed a formidable proposition since, for example, it would be nearly 13 times slower in execution time than a 9-state variable filter.

To make the 21-state variable filter practical for use in real time, the formulation of the engineering equations has been done by considering the number of numerical operations performed by the digital computer.

The detailed programing instructions will be presented in notation similar to FORTRAN. Namely, DØ loops will be used, not because of any expertise in programing, but because they offer the engineer an elegantly simple way to write large groups of equations. Also, the notation used will be understandable by the engineer; while, at the same time, it will be directly translatable into FORTRAN or machine language by the programer. Presentation of the programing instructions in this clearly understandable form makes it possible for the computer program to be implemented quickly with the least possible chance of error.

Variables used within the program may be categorized broadly into three types

1. Fixed point variables, used primarily as logic flags

2. Double-precision, floating-point, permanent-storage variables, used wherever quantities must be saved from one cycle to the next

3. Double-precision, floating-point, temporary-storage variables, used for quantities that need not be saved from one cycle to the next

The next three subsections will discuss these three types of variables in more detail. The definitions of these variables serve as a bridge between the engineering notation and the computer program notation.

14.1 The Fixed-Point Variables

There will be 48 fixed-point variables

$$F(L) \quad L = 1, 2, \ldots, 48$$

The first 35 of these variables will be defined here. The last 13 will be for future program use if the need should arise.

F(L) L = 1, 4

F(L) = 0, do not reset I_L, T_{OIL}, and $\epsilon_{\omega 3L}$

F(L) = 1, reset I_L and T_{OIL} (caused by bad data)

F(L) = -1, reset I_L, T_{OIL}, and $\epsilon_{\omega 3L}$ (caused by station location change)

F(L+4) L = 1, 4

F(L+4) = 0, current data bad or missing for station L

F(L+4) = 1, current data good for station L

F(L+8) L = 1, 4

F(L+8) = last value of F(L+4)

(Note: it may be desirable to display F(L+8).)

F(L+12) L = 1, 4

F(L+12) = 0 for 3-way data from station L

F(L+12) = 1 for 2-way data from station L

F(L+16) L = 1, 4 and F(21)

F(L+16) is the station L identification number; F(21) is the transmitter identification number

F(L+21) L = 1, 5

F(L+21) = the last value of F(L+16)

F(31)

F(31) = 0 for no good tracking data

F(31) = 1 if there is good tracking data

F(32)

F(32) = 0 for descent

F(32) = 1 for ascent from surface of the moon

F(33)

F(33) is the previous value of F(31)

F(34)

F(34) = 0 for free-flight before powered descent

F(34) = 1 otherwise

F(35)

F(35) = 0 under normal program operating conditions

F(35) = 1 under program restart conditions

14.2 Permanent Storage Variables

The following variables are double-precision, floating-point variables whose values must be preserved from one cycle to the next.

X(I) I = 1, 21

X(I) is the state vector, the elements of which were described in section 4

YY(I) I = 1, 4

The actual measurement vector, N^{*}_{FI}, described in section 8

JX(I, J) I = 1, 21 J = 1, 21

J(I, J) is the symmetric, state error-covariance matrix defined in section 3; J(I, J) I = 1, 13 J = 14, 21 is not used by the program. To avoid conflict with the integer J, the J(I, J) matrix will be designated as JX(I, J) in the program.

P(I) I = 1, 180

P(I) consists of input constants and program generated constants. The first 126 P's are defined below. The remaining 54 P's are reserved for future use by the program. If storage is a problem, the first 104 P's may be equivalenced to JX(I, J) I = 1, 13 J = 14, 21.

$$\begin{bmatrix} P(1) & P(2) & P(3) \\ P(4) & P(5) & P(6) \\ P(7) & P(8) & P(9) \end{bmatrix}$$ = the RNP matrix. $(RNP)^T$ is used to convert station location to MNBY coordinates.

P(10) = $\exp[-|\Delta T_O|/T_{\omega P}]$

P(11) = $\exp[-|\Delta T_O|/T_{\omega Y}]$

P(12) = $\exp[-|\Delta T_O|/T_{\epsilon \dot{M}}]$

P(13) = $\exp[-|\Delta T_O|/T_{\epsilon I}]$

P(14) = $\exp[-|\Delta T_O|/T_{\omega 3}]$

P(15) = T_F, time tag of state vector in the filter

P(16) = ΔT_F, filter integration step size

P(17) = ΔT_{MAX}, the maximum ΔT used by TRAJ

P(18) = T_O, current observation time

P(19) = ΔT_O, time interval between usable observations

P(20) = T_L, constant time lag between real time and T_0

P(21) = T_D, desired time of output state vector

P(22) = T_{DL}, the last value of T_D

P(23) = time tag for output from program

P(24) = $x_{V/M}$

P(25) = $y_{V/M}$

P(26) = $z_{V/M}$

P(27) = $\dot{x}_{V/M}$

P(28) = $\dot{y}_{V/M}$

P(29) = $\dot{z}_{V/M}$

LM position and velocity output from the program, with time tag of P(23), and in MNBY selenocentric coordinates

P(L+29) = longitude (+ east) of station L, L = 1, 4

P(34) = longitude of transmitter

P(L+34) = $R_{EL} \cos \varphi'_L + H_L \cos \varphi_L = \sqrt{(x'_{SL})^2 + (y'_{SL})^2}$, L = 1, 4

P(39) = $R_E \cos \varphi' + H \cos \varphi$ for transmitter

P(L+39) = $R_{EL} \sin \varphi'_L + H_L \sin \varphi_L = z'_{SL}$, L = 1, 4

P(44) = $R_E \sin \varphi' + H \sin \varphi$ for transmitter

P(45) = maximum allowable value of $b_L + \epsilon_{\omega 3L}$ before station L's data is deleted by the filter

P(46) = $\sigma^2_{\omega 3}$ for a 3-way station

P(47) = $\sigma^2_{\omega 3}$ for a 2-way station

P(L+47) = $\sigma^2_{\omega 3}$ for station L, L = 1, 4

P(L+51) = b_L, the a priori estimate of the station L "rate bias" error, L = 1, 4

P(L+55) = value for last cycle of N^*_{FL} = YY(L), L = 1, 4

$$\left.\begin{array}{l} P(60) = x_{M/E} \\ P(61) = y_{M/E} \\ P(62) = z_{M/E} \\ P(63) = \dot{x}_{M/E} \\ P(64) = \dot{y}_{M/E} \\ P(65) = \dot{z}_{M/E} \end{array}\right\}$$

position and velocity of the moon with respect to the earth in geocentric, MNBY coordinates

P(66) = μ_E, earth's gravitational constant

P(67) = μ_M, moon's gravitational constant

P(68) = $\mu_E + \mu_M$

P(69) = g in $gI\dot{M}/M$

P(70) = ω_{PN}, nominal value of $\dot{\psi}_P$

P(71) = ω_{YN}, nominal value of $\dot{\psi}_Y$

P(72) = $\dot{M}_N (\dot{M}_N \leq 0)$, nominal value of $\dot{M}$

P(73) = I_N, nominal value of specific impulse

$$\begin{bmatrix} P(74) & P(77) & P(80) \\ P(75) & P(78) & P(81) \\ P(76) & P(79) & P(82) \end{bmatrix} = \begin{bmatrix} c_1 & c_4 & c_7 \\ c_2 & c_5 & c_8 \\ c_3 & c_6 & c_9 \end{bmatrix}$$

$P(83) = 1 - [\exp(-|\Delta T_0|/T_{\omega P})]^2$

$P(84) = 1 - [\exp(-|\Delta T_0|/T_{\omega Y})]^2$

$P(85) = 1 - [\exp(-|\Delta T_0|/T_{\epsilon \dot{M}})]^2$

$P(86) = 1 - [\exp(-|\Delta T_0|/T_{\epsilon I})]^2$

$P(87) = 1 - [\exp(-|\Delta T_0|/T_{\omega 3})]^2$

$P(88) = \sigma^2_{\omega P}$

$P(89) = \sigma^2_{\omega Y}$

$P(90) = \sigma^2_{\epsilon \dot{M}}$

$P(91) = \sigma^2_{\epsilon I}$

$P(92) = \sigma^2_{ag} \;\; \Delta T_0{}^2$, velocity variance due to random gravitational acceleration

$P(93) = \dfrac{\omega_4 \; \nu_{tr}}{c}$, a measurement data constant

$P(94) = \omega_3$, biasing frequency for all stations

$P(95) = c$, the speed of light

$P(96) = \omega_{Earth}$, angular velocity of the earth

$P(L+96) = T_{OIL}$ for station L, L = 1, 4

$P(101) = \sigma^2_w$, variance of measurement noise

P(102) = initial value of J(I, I) I = 1, 3 for descent

P(103) = initial value of J(I, I) I = 4, 6 for descent

P(104) = initial value of $J(I, I)$ $I = 7, 8$ for descent.

P(105) = initial value of $J(I, I)$ $I = 9$ for descent

P(106) = initial value of $J(I, I)$ $I = 1, 3$ for ascent

P(107) = initial value of $J(I, I)$ $I = 4, 6$ for ascent

P(108) = initial value of $J(I, I)$ $I = 7, 8$ for ascent

P(109) = initial value of $J(I, I)$ $I = 9$ for ascent

P(L+109) = $\Delta YY_p(L)$ $L = 1, 4$, the past value of $\Delta YY(L)$

P(114) = Δ_{1MIN}, the minimum allowable value of $\Delta YY(L)$

P(115) = Δ_{1MAX}, the maximum allowable value of $\Delta YY(L)$

P(116) = Δ_{2MAX}, the maximum allowable value of $|\Delta YY(L)-\Delta YY_p(L)|$

P(117) = initial value of mass for descent

P(118) = initial value of mass for ascent

P(119) = approximate value of the transit time of electromagnetic waves from the spacecraft to the receiving station

P(120) = motor on time, in hours from midnight of launch day, for descent; <u>motor on time should be at the start of ullage</u>

P(121) = initial value of $\dot{M}_N$ for ascent

P(122) = powered flight value of $\sigma^2_{\epsilon\dot{M}}$, used when tracking data is available

P(123) = powered flight value of $\sigma^2_{\varepsilon\dot{M}}$, used when no tracking data is available

P(124) = time interval for processing free flight tracking data prior to powered descent initiation

P(125) = time offset to engine-on-time for the ascent phase

P(126) = time offset to engine-on-time for the descent phase

14.3 Temporary Storage Variables

All temporary storage variables are double precision. The sizes of the various matrices and vectors are shown here.

M(4, 21)	D(21, 4)	H(4, 4)	B(21, 4)
A(13, 13)	Y(4)	E(169)	

To save storage, the vector E(I) should be equivalenced to the 169 elements of A(I, J). Also, the matrix D(I, J) should be equivalenced to the first 84 elements of A(I, J), and B(I, J) should be equivalenced to the next 84 elements of A(I, J). However, note that the operation of the program is not dependent upon the equivalence of the above variables. The equivalencing is only a storage saving feature.

The total storage required by the variables mentioned in this section, and the preceding two sections, is 1782 words of core storage. Note that a single precision version of this program (practical only on a CDC machine) requires only 915 words of core storage.

14.4 The Subroutine TRANST Programing Instructions

In section 12, the engineering equations for the subroutine TRANST were discussed. The engineering equations are "linked" to the programing instructions by the definitions of the E cells shown here.

$E(1) = \omega_{Earth} \times T_0$

<u>For L = 1, 2, 3, 4</u>

$E(L+1) = x'_{SL}(T_0)$ $\quad$ $E(L+5) = y'_{SL}(T_0)$

$E(L+9) = x_{SL}(T_0)$ $\quad$ $E(L+13) = y_{SL}(T_0)$ $\quad$ $E(L+17) = z_{SL}(T_0)$

$E(L+21) = T_{VL}$

$E(L+25) = x(T_{VL})$ $\quad$ $E(L+29) = y(T_{VL})$ $\quad$ $E(L+33) = z(T_{VL})$

$E(L+37) = x(T_{VL}) - x_{SL}(T_0)$

$E(L+41) = y(T_{VL}) - y_{SL}(T_0)$

$E(L+45) = z(T_{VL}) - z_{SL}(T_0)$

$E(L+49) = |\underline{R}_{V/E}(T_{VL}) - \underline{R}_{SL}(T_0)|$

$E(54) = \hat{T}_T$ $\quad$ $E(55) = P(34) + \omega_E \times \hat{T}_T$

$E(56) = x'_T(\hat{T}_T)$ $\quad$ $E(57) = y'_T(\hat{T}_T)$

$E(58) = \dot{x}'_T(\hat{T}_T)$ $\quad$ $E(59) = \dot{y}'_T(\hat{T}_T)$

$E(60) = x_T(\hat{T}_T)$ $\quad$ $E(61) = y_T(\hat{T}_T)$ $\quad$ $E(62) = z_T(\hat{T}_T)$

$E(63) = \dot{x}_T(\hat{T}_T)$ $\quad$ $E(64) = \dot{y}_T(\hat{T}_T)$ $\quad$ $E(65) = \dot{z}_T(\hat{T}_T)$

<u>For L = 1, 2, 3 4</u>

$E(L+65) = T_{TL}$

$E(L+69) = x(T_{VL}) - x_T(T_{TL})$

$E(L+73) = y(T_{VL}) - y_T(T_{TL})$

$E(L+77) = z(T_{VL}) - z_T(T_{TL})$

$$E(L+81) = |\underline{R}_{V/E}(T_{VL}) - \underline{R}_T(T_{TL})|$$

$$E(L+85) = T_{VL} - T_F$$

$$E(L+89) = T_{TL} - \hat{T}_T$$

The detailed programing instructions are as follows

```
      E(1)  =  P(96)*P(18)
      DØ a  L = 1, 4
      E(L+1)  =  P(L+34)*DCØS(P(L+29) + E(1))
      E(L+5)  =  P(L+34)*DSIN(P(L+29) + E(1))
      E(L+9)  =  P(1)*E(L+1) + P(4)*E(L+5) + P(7)*P(L+39)
      E(L+13)  =  P(2)*E(L+1) + P(5)*E(L+5) + P(8)*P(L+39)
      E(L+17)  =  P(3)*E(L+1) + P(6)*E(L+5) + P(9)*P(L+39)
      E(L+21)  =  P(18) - (DSQRT((X(1) - E(L+9))**2
                + (X(2) - E(L+13))**2 + (X(3) - E(L+17))**2))/P(95)
      E(L+85)  =  E(L+21) - P(15)
      E(L+25)  =  X(1) + E(L+85)*X(4)
      E(L+29)  =  X(2) + E(L+85)*X(5)
      E(L+33)  =  X(3) + E(L+85)*X(6)
      E(L+37)  =  E(L+25) - E(L+9)

      E(L+41)  =  E(L+29) - E(L+13)
      E(L+45)  =  E(L+33) - E(L+17)
a     E(L+49)  =  DSQRT(E(L+37)**2 + E(L+41)**2 + E(L+45)**2)
      E(54)  =  E(22) - P(18) + E(22)
      E(55)  =  P(34) + P(96)*E(54)
```

```
     E(56)   =  P(39)*DCØS(E(55))
     E(57)   =  P(39)*DSIN(E(55))
     E(58)   =  -P(96)*E(57)
     E(59)   =  P(96)*E(56)
     DØ b    I = 1, 3
     E(I+59)  =  P(I)*E(56) + P(I+3)*E(57) + P(I+6)*P(44)
b    E(I+62)  =  P(I)*E(58) + P(I+3)*E(59)
     DØ c    L = 1, 4
     E(L+65)  =  E(L+21) - (DSQRT((E(L+25) - E(60))**2
                + (E(L+29) - E(61))**2 + (E(L+33) - E(62))**2))/P(95)
     E(L+89)  =  E(L+65) - E(54)
     E(L+69)  =  E(L+25) - E(60) - E(L+89)*E(63)
     E(L+73)  =  E(L+29) - E(61) - E(L+89)*E(64)
     E(L+77)  =  E(L+33) - E(62) - E(L+89)*E(65)
c    E(L+81)  =  DSQRT(E(L+69)**2 + E(L+73)**2 + E(L+77)**2)
```

Note that X(I), P(I), and E(I) should be in common.

14.5 The Subroutine TRAJ Programing Instructions

This subroutine will integrate the first six elements of the state vector, and the position and velocity of the moon over the interval ΔT seconds according to the equations outlined in section 5. The maximum step size is controlled by the logic shown in figure 6. At the present time, it is not anticipated that this option will ever be exercised. It is included only as a safety precaution to avoid an overly large integration step. This option is designed primarily for free-flight integration. If the option is exercised in powered flight, TRAJ assumes that ψ_P, ψ_Y and M are constant over the total integration step.

As shown here, the definitions of the E cells link the engineering equations to the programing instructions.

E(1) = time of the state estimate

$E(2) = x$	$E(3) = y$	$E(4) = z$
$E(5) = \dot{x}$	$E(6) = \dot{y}$	$E(7) = \dot{z}$
$E(8) = x_{M/E}$	$E(9) = y_{M/E}$	$E(10) = z_{M/E}$
$E(11) = \dot{x}_{M/E}$	$E(12) = \dot{y}_{M/E}$	$E(13) = \dot{z}_{M/E}$
$E(14) = x - x_{M/E}$	$E(15) = y - y_{M/E}$	$E(16) = z - z_{M/E}$
$E(17) = \omega_{PN} + \epsilon_{\omega P}$	$E(18) = \omega_{YN} + \epsilon_{\omega Y}$	$E(19) = \dot{M}_N + \epsilon_{\dot{M}}$
$E(20) = \sin \psi_P$	$E(21) = \sin \psi_Y$	
$E(22) = \cos \psi_P$	$E(23) = \cos \psi_Y$	
$E(24) = \sin \psi_Y \sin \psi_P$	$E(25) = \sin \psi_Y \cos \psi_P$	
$E(26) = \cos \psi_Y \sin \psi_P$	$E(27) = \cos \psi_Y \cos \psi_P$	
$E(28) = \lvert\underline{R}_{V/E}\rvert^2$	$E(29) = \lvert\underline{R}_{V/E}\rvert$	$E(30) = \mu_E/\lvert\underline{R}_{V/E}\rvert^3$
$E(31) = \Delta T$	$E(32) = \Delta T^2/2$	$E(33) = \Delta T^3/6$
$E(34) = \lvert\underline{R}_{M/E}\rvert^2$	$E(35) = \lvert\underline{R}_{M/E}\rvert$	$E(36) = \mu_M/\lvert\underline{R}_{M/E}\rvert^3$
$E(37) = \frac{g}{M}$	$E(38) = \frac{g}{M}(I_N+\epsilon_I)$	$E(39) = \frac{g}{M}(\dot{M}_N+\epsilon_{\dot{M}})$
$E(40) = \lvert\underline{R}_{V/E}-\underline{R}_{M/E}\rvert^2$	$E(41) = \lvert\underline{R}_{V/E}-\underline{R}_{M/E}\rvert$	$E(42) = \mu_M/\lvert\underline{R}_{V/E}-\underline{R}_{M/E}\rvert^3$

$E(43) = \frac{x}{|\underline{R}_{V/E}|}$ $\quad E(44) = \frac{y}{|\underline{R}_{V/E}|}$ $\quad E(45) = \frac{z}{|\underline{R}_{V/E}|}$

$E(46) = C_x$ $\quad E(47) = C_y$ $\quad E(48) = C_z$

$E(49) = \frac{x_{M/E}}{|\underline{R}_{M/E}|}$ $\quad E(50) = \frac{y_{M/E}}{|\underline{R}_{M/E}|}$ $\quad E(51) = \frac{z_{M/E}}{|\underline{R}_{M/E}|}$

$E(52) = \frac{\mu_E+\mu_M}{|\underline{R}_{M/E}|^3}$ $\quad E(53) = \frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} + \frac{\mu_E}{|\underline{R}_{V/E}|^3}$

$E(54) = \frac{-\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} + \frac{\mu_M}{|\underline{R}_{M/E}|^3}$

$E(55) = \frac{x - x_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$ $\quad E(56) = \frac{y - y_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$ $\quad E(57) = \frac{z - z_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$

$E(58) = \frac{g}{M}(I_N+\epsilon_I)(\dot{M}_N+\epsilon_{\dot{M}})$ $\quad E(59) = \frac{g}{M}(I_N+\epsilon_I)(\dot{M}_N+\epsilon_{\dot{M}})\frac{1}{M}$

$E(60) = \frac{\partial\ddot{x}}{\partial x}$ $\quad E(61) = \frac{\partial\ddot{x}}{\partial y}$ $\quad E(62) = \frac{\partial\ddot{x}}{\partial z}$

$E(63) = \frac{\partial\ddot{x}}{\partial x_{M/E}}$ $\quad E(64) = \frac{\partial\ddot{x}}{\partial y_{M/E}}$ $\quad E(65) = \frac{\partial\ddot{x}}{\partial z_{M/E}}$

$E(66) = \frac{\partial\ddot{x}}{\partial\psi_P}$ $\quad E(67) = \frac{\partial\ddot{x}}{\partial\psi_Y}$ $\quad E(68) = \frac{\partial\ddot{x}}{\partial M}$

$E(69) = \frac{\partial\ddot{x}}{\partial\epsilon_{\dot{M}}}$ $\quad E(70) = \frac{\partial\ddot{x}}{\partial\epsilon_I}$

$E(71) = \frac{\partial\ddot{y}}{\partial x}$ $\quad E(72) = \frac{\partial\ddot{y}}{\partial y}$ $\quad E(73) = \frac{\partial\ddot{y}}{\partial z}$

$E(74) = \frac{\partial\ddot{y}}{\partial x_{M/E}}$ $\quad E(75) = \frac{\partial\ddot{y}}{\partial y_{M/E}}$ $\quad E(76) = \frac{\partial\ddot{y}}{\partial z_{M/E}}$

$$E(77) = \frac{\partial \ddot{y}}{\partial \psi_P} \qquad E(78) = \frac{\partial \ddot{y}}{\partial \psi_Y} \qquad E(79) = \frac{\partial \ddot{y}}{\partial M}$$

$$E(80) = \frac{\partial \ddot{y}}{\partial \epsilon_{\dot{M}}} \qquad E(81) = \frac{\partial \ddot{y}}{\partial \epsilon_I}$$

$$E(82) = \frac{\partial \ddot{z}}{\partial x} \qquad E(83) = \frac{\partial \ddot{z}}{\partial y} \qquad E(84) = \frac{\partial \ddot{z}}{\partial z}$$

$$E(85) = \frac{\partial \ddot{z}}{\partial x_{M/E}} \qquad E(86) = \frac{\partial \ddot{z}}{\partial y_{M/E}} \qquad E(87) = \frac{\partial \ddot{z}}{\partial z_{M/E}}$$

$$E(88) = \frac{\partial \ddot{z}}{\partial \psi_P} \qquad E(89) = \frac{\partial \ddot{z}}{\partial \psi_Y} \qquad E(90) = \frac{\partial \ddot{z}}{\partial M}$$

$$E(91) = \frac{\partial \ddot{z}}{\partial \epsilon_{\dot{M}}} \qquad E(92) = \frac{\partial \ddot{z}}{\partial \epsilon_I}$$

$$E(93) = \ddot{x} \qquad E(94) = \ddot{y} \qquad E(95) = \ddot{z}$$

$$E(96) = \dddot{x} \qquad E(97) = \dddot{y} \qquad E(98) = \dddot{z}$$

$$E(99) = N$$

$$E(100) = 3\,\frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3}\;\frac{x - x_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$$

$$E(101) = 3\,\frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3}\;\frac{y - y_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$$

$$E(102) = 3\,\frac{\mu_E}{|\underline{R}_{V/E}|^3}\;\frac{x}{|\underline{R}_{V/E}|} \qquad E(103) = 3\,\frac{\mu_E}{|\underline{R}_{V/E}|^3}\;\frac{y}{|\underline{R}_{V/E}|}$$

$$E(104) = 3\,\frac{\mu_M}{|\underline{R}_{M/E}|^3}\;\frac{x_{M/E}}{|\underline{R}_{M/E}|} \qquad E(105) = 3\,\frac{\mu_M}{|\underline{R}_{M/E}|^3}\;\frac{y_{M/E}}{|\underline{R}_{M/E}|}$$

The detailed programing instructions are shown here. Note that X(I), P(I), and E(I) should be in common.

```
      E(20)  =  DSIN(X(7))
      E(21)  =  DSIN(X(8))
      E(22)  =  DCØS(X(7))
      E(23)  =  DCØS(X(8))
      E(24)  =  E(21)*E(20)
      E(25)  =  E(21)*E(22)
      E(26)  =  E(23)*E(20)
      E(27)  =  E(23)*E(22)
      IF |E(31)| > P(17) GØ TØ a
      N  =  1
      GØ TØ b
a     N = 1. DO + (DABS(E(31)))/P(17)
      E(99)  =  N
      E(31)  =  E(31)/E(99)
b     DØ n    L = 1, N
      E(1)  =  E(1) + E(31)
      DØ c   I = 1, 3
      E(I+13)  =  E(I+1) - E(I+7)
c     E(I+16)  =  P(I+69) + X(I+9)
      DØ d   I = 1, 13, 6
      E(I+27)  =  E(I+1)**2 + E(I+2)**2 + E(I+3)**2
```

```
d     E(I+28)  =  DSQRT(E(I+27))
      E(32)  =  E(31)**2/2.DO
      E(33)  =  E(32)*E(31)/3.DO
      E(30)  =  (P(66)/E(28))/E(29)
      E(36)  =  (P(67)/E(34))/E(35)
      E(42)  =  (P(67)/E(40))/E(41)
      E(37)  =  P(69)/X(9)
      E(38)  =  E(37)*(P(73) + X(13))
      E(39)  =  E(37)*E(19)
      DØ e   J = 1, 3
      E(J+45)  =  P(J+73)*E(26) + P(J+76)*E(27) + P(J+79)*E(21)
      DØ e   I = 1, 13, 6
      K  =  I+J
e     E(K+41)  =  E(K)/E(I+28)
      E(52)  =  (P(68)/E(34))/E(35)
      E(53)  =  E(42) + E(30)
      E(54)  =  -E(42) + E(36)
      E(58)  =  E(38)*E(19)
      E(59)  =  E(58)/X(9)
      DØ f   I = 1, 2
      E(I+99)  =  3. DO*E(42)*E(I+54)
      E(I+101) =  3. DO*E(30)*E(I+42)
```

```
f    E(I+103) =  3. DO*E(36)*E(I+48)
     DØ g  I = 1, 3
     E(I+59)  =  E(100)*E(I+54) + E(102)*E(I+42)
g    E(I+62)  =  -E(100)*E(I+54) -E(104)*E(I+48)
     DØ h  I = 1, 2
     E(I+71)  =  E(101)*E(I+55) + E(103)*E(I+43)
h    E(I+74)  =  -E(101)*E(I+55) -E(105)*E(I+49)
     E(71)  =  E(61)
     E(74)  =  E(64)
     DØ i  I = 1, 2
     J  = 57 + 3*I
     E(J)  =  E(J) - E(I+52)
     E(J+12)  =  E(J+12) - E(I+52)
     E(J+22)  =  E(J+2)
     E(J+23)  =  E(J+13)
i    E(J+24)  =  -E(J) - E(J+12)
     DØ j  I = 1, 3
     J  =  55 + 11*I
     E(J)  =  E(58)*(-P(I+73)*E(27) + P(I+76)*E(26))
     E(J+1)  =  E(58)*(P(I+73)*E(24) + P(I+76)*E(25) -P(I+79)*E(23))
     E(J+2)  =  E(I+45)*E(59)
     E(J+3)  =  -E(I+45)*E(38)
```

```
j     E(J+4)  =  -E(I+45)*E(39)
      DØ k   I = 1, 3
      J  =  49 + 11*I
      E(I+92)  =  -E(42)*E(I+13) -E(30)*E(I+1) -E(36)*E(I+7) -E(I+45)*E(58)
      E(I+95)  =  E(J)*E(5) + E(J+1)*E(6) + E(J+2)*E(7) + E(J+3)*E(11)
                 + E(J+4)*E(12) + E(J+5)*E(13) + E(J+6)*E(17)
                 + E(J+7)*E(18) + E(J+8)*E(19)
k     E(I+1)  =  E(I+1) + E(I+4)*E(31) + E(I+92)*E(32) + E(I+95)*E(33)
      DØ m   I = 1, 3
      E(I+10)  =  E(I+10) -E(52)*E(I+7)*E(31)
      E(I+7)  =  E(I+7)*(1.D0 + E(52)*E(32)) + E(I+10)*E(31)
m     E(I+4)  =  E(I+4) + E(I+92)*E(31) + E(I+95)*E(32)
n     CØNTINUE
```

14.6 Programing Instructions for Blocks 1 and 2

Blocks 1 and 2 are the first two blocks in the overall block diagram shown in figure 5. The programing instructions for these two blocks are shown here. Based on results from future simulation studies, the equations in block 1 may have to be slightly modified. The logic for block 1 is included in flow chart 1 of the appendix.

comment BEGIN BLOCK 1

Zero F(I), X(I), YY(I), JX(I, J), P(I), Y(I)

Input the nonzero P(I) input constants. (If all P(I), including zero values, are read in, then omit zeroing P(I) above.)

Set descent (F(32) = 0), ascent (F(32) = 1) flag.

Read in the RNP matrix as shown below:

$$\begin{bmatrix} P(1) & P(2) & P(3) \\ P(4) & P(5) & P(6) \\ P(7) & P(8) & P(9) \end{bmatrix} = RNP$$

Obtain time of powered flight initiation from Mission Plan Table (MPT) and store in P(120)

comment INITIALIZE FOR DESCENT OR ASCENT

IF (F(32) $\neq$ 0) GO TO yy

P(15) = P(120) - P(124)

IF((T_{RL} - P(20)). LT. P(15)) GO TO xx

IF ((T_{RL} - P(20)). LT. P(120)) GO TO ww

P(15) = P(120)

P(18) = $(T_{RL} - P(20))_R$ [a]

STORE MPT LM MCI VECTOR IN P(I), I = 60, 65

GO TO zz

ww P(15) = T_{RL} - P(20)

xx P(18) = $(P(15) + P(119))_R$ [a]

[a] This step, though easily done, is very critical. If not done properly, the filter may get out of "sync" with the observation times and fail to take in any measurements. Note, also, that it is assumed that measurements are available exactly on the second mark. For example, 5-measurements-per-second data cannot be processed with time tags of 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, ... seconds.

```
        CALL AEG. INTEGRATE MPT LM MCI VECTOR TO P(15).
          STORE IN P(I), I = 60, 65.

        GO TO zz

yy      P(15) = P(120) - P(125)

        COMPUTE MCT STATE VECTOR OF LAUNCH SITE.  (See flow chart 1
          in appendix.)
        CALL ELVCNV.  CONVERT MCT VECTOR TO MCI AND STORE MCI VECTOR
          IN P(I), I = 60, 65.
        OBTAIN CSM MCI STATE VECTOR AT P(15) FROM VEHICLE EPHEMERIS.
          STORE IN P(I), I = 74, 79.

zz      CALL ELVCNV.  CONVERT LM MCI VECTOR TO ECI.  STORE ECI VECTOR
          IN X(I), I = 1, 6.

        IF F(32) ≠ 0  GØ TØ b

        DØ a  I = 1, 3

        E(I)  =  P(I+59)

        E(I+3)  =  -P(I+62)

        JX(I, I)  =  P(102)

a       JX(I+3, I+3)  =  P(103)

        JX(7, 7)  =  P(104)

        JX(8, 8)  =  P(104)

        JX(9, 9)  =  P(105)

        X(9)  =  P(117)

        GØ TØ d
```

```
b        DØ c   I = 1, 3
         E(I)  =  P(I+73)
         E(I+3)  =  P(I+76)
         JX(I, I)  =  P(106)
c        JX(I+3, I+3)  =  P(107)
         JX(7, 7)  =  P(108)
         JX(8, 8)  =  P(108)
         JX(9, 9)  =  P(109)
         JX(12, 12)  =  P(123)
         P(90)  =  P(123)
         P(72)  =  P(121)
         F(34)  =  1
         X(7)  =  1.5625 DO
         X(9)  =  P(118)
d        JX(10, 10)  =  P(88)
         JX(11, 11)  =  P(89)
         JX(13, 13)  =  P(91)
comment  CALCULATE C MATRIX
         E(7)  =  DSQRT(P(60)**2 + P(61)**2 + P(62)**2)
         DØ e   I = 1, 3
e        P(I + 73)  =  P(I+59)/E(7)
         E(8)  =  E(2)*E(6) -E(3)*E(5)
         E(9)  =  E(3)*E(4) -E(1)*E(6)
```

```
         E(10)   =  E(1)*E(5) -E(2)*E(4)
         P(77)   =  E(9)*P(76) -E(10)*P(75)
         P(78)   =  E(10)*P(74) -E(8)*P(76)
         P(79)   =  E(8)*P(75) -E(9)*P(74)
         E(11)   =  DSQRT(P(77)**2 + P(78)**2 + P(79)**2)
         DØ f  I = 77, 79
f        P(I)  =  P(I)/E(11)
         P(80)   =  P(75)*P(79) -P(76)*P(78)
         P(81)   =  P(76)*P(77) -P(74)*P(79)
         P(82)   =  P(74)*P(78) -P(75)*P(77)
         DØ g  I = 1, 6
g        P(I+59)  =  X(I) -P(I+59)
comment  BEGIN BLOCK 2
         CALL TRANST
         P(16) = E(86)
```

14.7 Programing Instructions for Blocks 3, 4, 5, and 6

These blocks constitute the dynamics portion of the program. Based on results from future simulation studies, the equations in block 3 may have to be slightly modified. The logic for a restart procedure and the logic for block 3 are included in flow charts 2 and 3 respectively of the appendix.

Just before the instruction "IF P(18) + P(20) > current time WAIT" is the ideal time to interrupt this program to perform other shared operations with the computer. This instruction forces the program to be synchronized with real time. Because the program cycle will execute "faster" than real time, there always will be a pause at this instruction to allow real time to catch up.

```
a        CØNTINUE
         IF (P(18) + P(20)) > T_RL (current time in hours) WAIT
comment  BEGIN BLOCK 3
         IF (F(35) ≠ 0) GØ TØ aaa
         IF F(34) ≠ 0  GØ TØ ab
         IF P(15) < P(120)  GØ TØ ag
comment  INITIALIZE THRUST ANGLE FØR DESCENT
         DØ aa  I = 1, 3
aa       E(I)  =  P(I+62) - X(I+3)
aaa      F(35) = 0
         E(4)  =  P(74)*E(1) + P(75)*E(2) + P(76)*E(3)
         E(5)  =  P(77)*E(1) + P(78)*E(2) + P(79)*E(3)
         E(6)  =  P(80)*E(1) + P(81)*E(2) + P(82)*E(3)
         E(7)  =  DSQRT(E(4)**2 + E(5)**2)
         X(7)  =  DATAN2(E(4), E(5))
         X(8)  =  DATAN2(E(6), E(7))
         JX(12, 12)  =  P(123)
         P(90)  =  P(123)
         F(34)  =  1
comment  SET DATA DRØPØUT FLAG
ab       F(31)  =  1
         IF (F(9) + F(10) + F(11) + F(12)  =  0)  SET F(31)  =  0
comment  SET MASS FLØW RATE VARIANCE
         IF (F(31) - F(33))  ac, af, ad
```

```
ac        P(90)  =  P(123)
          GØ TØ ae
ad        JX(12, 12)  =  P(122)
          P(90)  =  P(122)
ae        F(33)  =  F(31)
comment   SET NØMINAL VALUE OF MDØT TØ BEST ESTIMATE ØF MDØT
af        P(72)  =  P(72) + X(12)
          X(12)  =  0. DO
ag        CØNTINUE
comment   BEGIN BLOCK 4
          E(31)  =  P(16)
          E(1)  =  P(15)
          DØ b   I = 1, 6
          E(I+1)  =  X(I)
b         E(I+7)  =  P(I+59)
          CALL TRAJ
          P(15)  =  E(1)
          DØ c   I = 1, 6
          X(I)  =  E(I+1)
c         P(I+59)  =  E(I+7)
comment   SET ΔT*
          IF F(34)  =  0  GØ TØ d
```

```
          M(4, 1)  =  P(16)
          GØ TØ e
d         M(4; 1)  =  0. DO
e         DØ f   I = 7, 9
f         X(I)  =  X(I) + E(I+10)*M(4, 1)
          DØ g   I = 10, 13
          X(I)  =  X(I)*P(I)
g         X(I+4)  =  X(I+4)*P(14)
comment   BEGIN BLOCK 5
          DØ h   J = 1, 3
          K  =  48 + 11*J
          DØ h   I = 1, 3
          K  =  K+1
          M(J, I)  =  E(K)*P(16)
          M(J, I+6)  =  E(K+6)*P(16)
h         M(J, I+11)  =  E(K+9)*P(16)
comment   BEGIN BLOCK 6
          DØ k   I = 1, 13
          DØ i   J = 1, 3
          A(I, J)  =  JX(I, J) + JX(I, J+3)*P(16)
i         A(I, J+3)  =  JX(I, 1)*M(J, 1) + JX(I, 2)*M(J, 2)
                      + JX(I, 3)*M(J,3) + JX(I, 7)*M(J, 7)
                      + JX(I, 8)*M(J,8) + JX(I, 9)*M(J, 9)
                      + JX(I, 12)*M(J, 12) + JX(I, 13)*M(J, 13)
                      + JX(I, J+3)
```

```
          DØ j   J = 7, 9
j         A(I, J)  =  JX(I, J) + JX(I, J+3)*M(4,1)
          DØ k   J = 10, 13
k         A(I, J)  =  JX(I, J)*P(J)
          DØ n   I = 1, 3
          DØ m   J = I, 13
m         JX(I, J)  =  A(I, J) + A(I+3, J)*P(16)
          L  =  I+ 3
          DØ n   J = L, 13
n         JX(L, J)  =  M(I, 1)*A(1, J) + M(I, 2)*A(2, J) + M(I,3)*A(3, J)
                     + M(I, 7)*A(7, J) + M(I, 8)*A(8, J) + M(I, 9)*A(9, J)
                     + M(I, 12)*A(12, J) + M(I, 13)*A(13, J) + A(L, J)
          DØ o   I = 7, 9
          DØ o   J = I, 13
o         JX(I, J)  =  A(I, J) + A(I+3, J)*M(4, 1)
          DØ p   I = 10, 13
          DØ p   J = I, 13
p         JX(I, J)  =  P(I)*A(I, J)
          DØ q   I = 2, 13
          L  =  I-1
          DØ q   J = 1, L
q         JX(I, J)  =  JX(J, I)
          DØ r   J = 1, 13
          DØ r   I = 1, 4
          A(I, J)  =  P(14)*JX(I+13, J)
```

```
r       A(I+4, J)  =  JX(I+17, J)
        DØ u   I = 1, 8
        DØ s   J = 1, 3
        JX(I+13, J)  =  A(I, J) + A(I, J+3)*P(16)
s       JX(I+13, J+3)  =  A(I, 1)*M(J, 1) + A(I, 2)*M(J, 2)
                        + A(I, 3)*M(J, 3) + A(I, 7)*M(J, 7)
                        + A(I, 8)*M(J, 8) + A(I, 9)*M(J, 9)
                        + A(I, 12)*M(J, 12) + A(I, 13)*M(J, 13)
                        + A(I, J+3)
        DØ t   J = 7, 9
t       JX(I+13, J)  =  A(I, J) + A(I, J+3)*M(4, 1)
        DØ u   J = 10, 13
u       JX(I+13, J)  =  A(I, J)*P(J)
        E(1)  =  P(14)**2
        DØ v   I = 14, 17
        DØ v   J = I, 17
        JX(I, J)  =  E(1)*JX(I, J)
v       JX(J, I)  =  JX(I, J)
        DØ w   I = 18, 21
        DØ w   J = 14, 17
        JX(I, J)  =  P(14)*JX(I, J)
w       JX(J, I)  =  JX(I, J)
comment ADD R
        DØ x   I = 4, 6
```

```
x       JX(I, I)  =  JX(I, I) + P(92)
        DØ y   I = 10, 13
        JX(I, I)  =  JX(I, I) + P(I+73)*P(I+78)
y       JX(I+4, I+4)  =  JX(I+4, I+4) + P(87)*P(I+38)
```

14.8 Programing Instructions for Blocks 7 through 14

The check on |F(1)| + |F(2)| + |F(3)| + |F(4)|, shown functionally between blocks 9 and 10 in the overall block diagram, is made by equivalent checks at the beginning of block 10.

```
comment  BEGIN BLOCK 7

         M = 0

         DØ g   L = 1, 4

         IF there is a good[a] cycle count for the Lth station in the
         interval P(18) -1.D-5 to P(18) + 1.D-5 hours GØ TØ a

         P(L+109) = 0.D0

         M = M+1

         F(L+4)  =  0

         F(L)  =  0

         GØ TØ f
```

[a]The word good means that the tracking station has attached a good label to the data.

```
a       Store count in YY(L)
        Set F(L+12)  =  0 for 3-way Doppler.[b]
        Set F(L+12)  =  1 for 2-way Doppler.[b]
        Load station ID number into F(L+16).[b]
        Load b_L, a priori "rate bias" in cycles/hour, into P(L+51).

        E(94)  =  YY(L) -P(L+55)
        IF (E(94) ≤ P(114)) GØ TØ b
        IF (E(94) ≥ P(115)) GØ TØ b
        IF (DABS(E(94) -P(L+109))) ≥ P(116)  GØ TØ b
        F(L+4)  =  1
        GØ TØ c
b       F(L+4)  =  0
c       P(L+109)  =  E(94)
        IF (F(L+4) -F(L+8)) > 0  GØ TØ d
        F(L)  =  0
        GØ TØ e
d       F(L)  =  1
e       IF F(L+16)  =  F(L+21)  GØ TØ f
```

[b]The information concerning the flag, F(L + 12), and the station ID number, F(L + 16), is contained in the tracking station input word.

Load geodetic longitude of station F(L+16) into P(L+29).[a]

Load R_E cos φ' + H cos φ for station F(L+16) into P(L+34).[a]

Load R_E sin φ' + H sin φ for station F(L+16) into P(L+39).[a]

```
          F(L)  =  -1
f         P(L+55)  =  YY(L)
g         F(L+8)  =  F(L+4)
          IF M = 4 GØ TØ i
```

Load transmitter ID into F(21).

```
          IF F(21)  =  F(26)  GØ TØ i
```

Load geodetic longitude of station F(21) into P(34).*

Load R_E cos φ' + H cos φ for station F(21) into P(39).*

Load R_E sin φ' + H sin φ for station F(21) into P(44).*

```
          DØ h   I = 1, 4
h         F(I)  =  -1
i         DØ j   I = 1, 5
j         F(I+21)  =  F(I+16)
comment   BEGIN BLOCK 8
          CALL TRANST
          P(16)  =  E(86) + P(19)
```

[a]These quantities may be loaded directly from the station characteristics table.

```
comment  BEGIN BLOCK 9
         DØ o   L = 1, 4
         IF F(L+4)  =  0   GØ TØ m
         K  =  L+33
         M(L, L+13)  =  P(18) -P(L+96)
         M(L, L+17)  =  -1.DO
         DØ k   J = 1, 3
         K  =  K+4
         M(L, J)  =  P(93)*(E(K)/E(L+49) + E(K+32)/E(L+81))
k        M(L, J+3)  =  E(L+85)*M(L, J)
         GØ TØ o
m        DØ n   I = 1, 6
n        M(L, I)  =  0. DO
         M(L, L+13)  =  0. DO
         M(L, L+17)  =  0. DO
o        CONTINUE
comment  BEGIN BLOCK 10
         DØ v   L = 1, 4
         IF (F(L))  =  0   GØ TØ v
         Y(L+17)  =  P(93)*(E(L+49) + E(L+81)) -YY(L)
         P(L+96)  =  P(18)
         DØ p   J = 1, 21
p        JX(L+17, J)  =  0. DO
         DØ q   J = 14, 21
```

```
q       JX(J, L+17)  =  0. DO
        IF (F(L)) > 0   GØ TØ v
        IF (F(L+12))  =  0  GØ TØ r
        P(L+47)  =  P(47)
        GØ TØ s
r       P(L+47)  =  P(46)
s       X(L+13)  =  0. DO
        DØ t   J = 1, 21
t       JX(L+13, J)  =  0. DO
        DØ u   J = 14, 21
u       JX(J, L+13)  =  0. DO
        JX(L+13, L+13)  =  P(L+47)
v       CØNTINUE
        DØ z   L = 1, 4
        IF (F(L))  =  0  GØ TØ z
        DØ w   J = 1, 21
w       JX(L+17, J)  =  M(L, 1)*JX(J, 1) + M(L, 2)*JX(J, 2)
                   + M(L, 3)*JX(J, 3) + M(L, 4)*JX(J, 4)
                   + M(L, 5)*JX(J, 5) + M(L, 6)*JX(J, 6)
        DØ x   J = 14, 21
x       JX(J, L+17)  =  JX(L+17, J)
```

```
         JX(L+17, L+17)  =  JX(L+17, L+17) + P(101)
         M(L, L+13)  =  0. D0
         M(L, L+17)  =  0. D0
         DØ y   J = 1, 6
y        M(L, J)  =  0. D0
z        CØNTINUE
comment  BEGIN BLOCK 11
         DØ aa   L = 1, 4
         Y(L)  = YY(L) -P(93)*(E(L+49) + E(L+81)) -(P(94) + P(L+51)
                + X(L+13))*(P(18) -P(L+96)) + X(L+17)
         IF (F(L+4)) = 0  SET Y(L)  = 0. D0
aa       CØNTINUE
         P(18)  =  P(18) + P(19)
         M  =  0
comment  BEGIN BLOCK 12
comment  FORM D = J(M TRANSPOSE)
ab       DØ ac   I = 1, 21
         DØ ac   J = 1, 4
ac       D(I, J)  =  JX(I, 1)*M(J, 1) + JX(I, 2)*M(J, 2) + JX(I, 3)*M(J,3)
                   + JX(I, 4)*M(J, 4) + JX(I, 5)*M(J, 5) + JX(I, 6)*M(J,6)
                   + JX(J+13, I)*M(J,J+13) + JX(J+17,I)*M(J,J+17)
comment  FORM H = MD + W
         DØ ae   I = 1, 4
         DØ ad   J = 1, I
```

```
ad        H(I, J)  =  M(I, 1)*D(1, J) + M(I, 2)*D(2, J) + M(I, 3)*D(3, J)
                    + M(I, 4)*D(4, J) + M(I, 5)*D(5, J) + M(I, 6)*D(6, J)
                    + M(I, I+13)*D(I+13, J) + M(I, I+17)*D(I+17, J)
ae        H(I, I)  =  H(I, I) + P(101)
          DØ aea   L = 1, 4
aea       M(L, 10)  =  H(L, L)*P(45)
comment   FORM H  =  H INVERSE
          H(1, 1)  =  1.DO/H(1, 1)
          DØ aj   J = 1, 3
          L  =  J+1
          DØ af   I = 1,J
          H(I, L)  =  0. DO
          DØ af   K = 1, J
af        H(I, L)  =  H(I, L) - H(L, K)*H(I, K)
          DØ ag   K = 1, J
ag        H(L, L)  =  H(L, L) + H(L, K)*H(K, L)
          H(L, L)  =  1.DO/H(L, L)
          DØ ah   I = 1, J
ah        H(L, I)  =  H(I, L)*H(L, L)
          DØ ai   I = 1, J
          DØ ai   K = I, J
          H(I, K)  =  H(I, K) + H(I, L)*H(L, K)
ai        H(K, I)  =  H(I, K)
          DØ aj   I = 1, J
aj        H(I, L)  =  H(L, I)
```

```
comment   FORM B = DH
          DØ ak   I = 1, 21
          DØ ak   J = 1, 4
ak        B(I, J)  =  D(I, 1)*H(1, J) + D(I, 2)*H(2, J) + D(I, 3)*H(3, J)
                    + D(I, 4)*H(4, J)
          IF M ≠ 0  GØ TØ ap

comment   BEGIN BLOCK 13
          DØ an   L = 1, 4
          IF (F(L+8))  =  0  GØ TØ an
          K  =  L+13
          H(L, 1)  =  Y(L)**2
          IF (H(L, 1) < M(L, 10) .AND. STATION IN AVERAGE)
                    GØ TØ an
          F(L+8)  =  0
          DØ am   J = 1, 6
am        M(L, J)  =  0. DO
          M(L, K)  =  0. DO
          M(L, L+17)  =  0. DO
an        CØNTINUE
          DØ ao   L = 1, 4
          IF (F(L+4)  =  0) GØ TØ ao
          IF (H(L, 1) < M(L, 10) .AND. STATION IN AVERAGE)
                    GØ TØ ao
          Y(L)  =  0.  DO
```

$$M = \left[\begin{array}{cccccccccccccccccccccc} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{1,14} & 0 & 0 & 0 & m_{1,18} & 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{2,15} & 0 & 0 & 0 & m_{2,19} & 0 & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{3,16} & 0 & 0 & 0 & m_{3,20} & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{4,17} & 0 & 0 & 0 & m_{4,21} \end{array}\right]$$

If the measurement flag, $F(L+4)$ $L = 1,4$, indicates that the data for station L is either bad or missing, then the Lth row of the M matrix is set to zero. This causes the state error covariance matrix to reflect the fact that the station L data is not being used.

Note that the error in calculating I_L, e_{IL}, is

$$e_{IL} = M(L,\ 1)\ e_x + M(L,\ 2)\ e_y + M(L,\ 3)\ e_z$$

$$+ M(L,4)\ e_{\dot{x}} + M(L,5)\ e_{\dot{y}} + M(L,6)\ e_{\dot{z}} - w_L$$

where e_x, e_y, ..., $e_{\dot{z}}$ are the errors in the filter's estimates of x, y, ..., $\dot{z}$. This equation will be used to reset the (L+13)th row and column of the state error covariance matrix whenever I_L is reinitialized. For example, let L = 2. Then

$$J(19,1) = E[e_{I2}\ e_x] = M(2,1)\ J(1,1) + M(2,2)\ J(1,2) + M(2,3)\ J(1,3)$$

$$+ M(2,4)\ J(1,4) + M(2,5)\ J(1,5) + M(2,6)\ J(1,6)$$

and

$$J(19,2) = E[e_{I2}\, e_y] = M(2,1)\, J(2,1) + M(2,2)\, J(2,2) + M(2,3)\, J(2,3)$$

$$+ M(2,4)\, J(2,4) + M(2,5)\, J(2,5) + M(2,6)\, J(2,6)$$

and so on.

9.0 DATA INPUT LOGIC AND DATA STATUS CHECKS

The following integers are defined to assist in the handling of the measurements.

<u>F(L) L = 1, 2, 3, 4</u>

F(L) = 0 do not reset I_L, T_{OIL}, and $\varepsilon_{\omega 3L}$

F(L) = 1 reset I_L and T_{OIL}. (Caused by bad data.)

F(L) = -1 reset I_L, T_{OIL}, and $\varepsilon_{\omega 3L}$. (Caused by station location change.)

<u>F(L+4) L = 1, 2, 3, 4</u>

F(L+4) = 0, current data bad or missing for station L

F(L+4) = 1, current data good for station L

<u>F(L+8) L = 1, 2, 3, 4</u>

F(L+8) = last value of F(L+4). Note: it may be desirable to display these flags.

<u>F(L+12) L = 1, 2, 3, 4</u>

F(L+12) = 0 for 3-way data from station L

F(L+12) = 1 for 2-way data from station L

<u>F(L+16) L = 1, 2, 3, 4</u>

F(L+16) is the station L identification number

F(21)

F(21) is the transmitter identification number

F(L+21)   L = 1, 2, 3, 4, 5

F(L+21) = the last value of F(L+16)

Part of the data status checks involves editing of the data. A simple, but effective, data editing scheme is used to tag the data good or bad. The scheme is a "three-point" editing procedure which checks the current data against the two previous data points. To be labeled good, the current data point must pass both a gross reasonableness test and a smoothness check. The gross reasonableness test is made by checking the slope $(N_0 - N_1)/\Delta T_0$, (fig. 2). The slope must lie within reasonable bounds; otherwise, the cycle count at T_0 will be labeled bad. For example, the slope is approximated as

$$(N_0 - N_1)/\Delta T_0 \approx \omega_3 + 2\,\frac{\omega_4\, \nu_{tr}}{c} \text{ (velocity along the line of sight)}$$

where

$$\omega_3 = 10^6 \text{ cycles/sec} \qquad 2\,\frac{\omega_4\, \nu_{tr}}{c} = 4.64 \text{ cycles/ft}$$

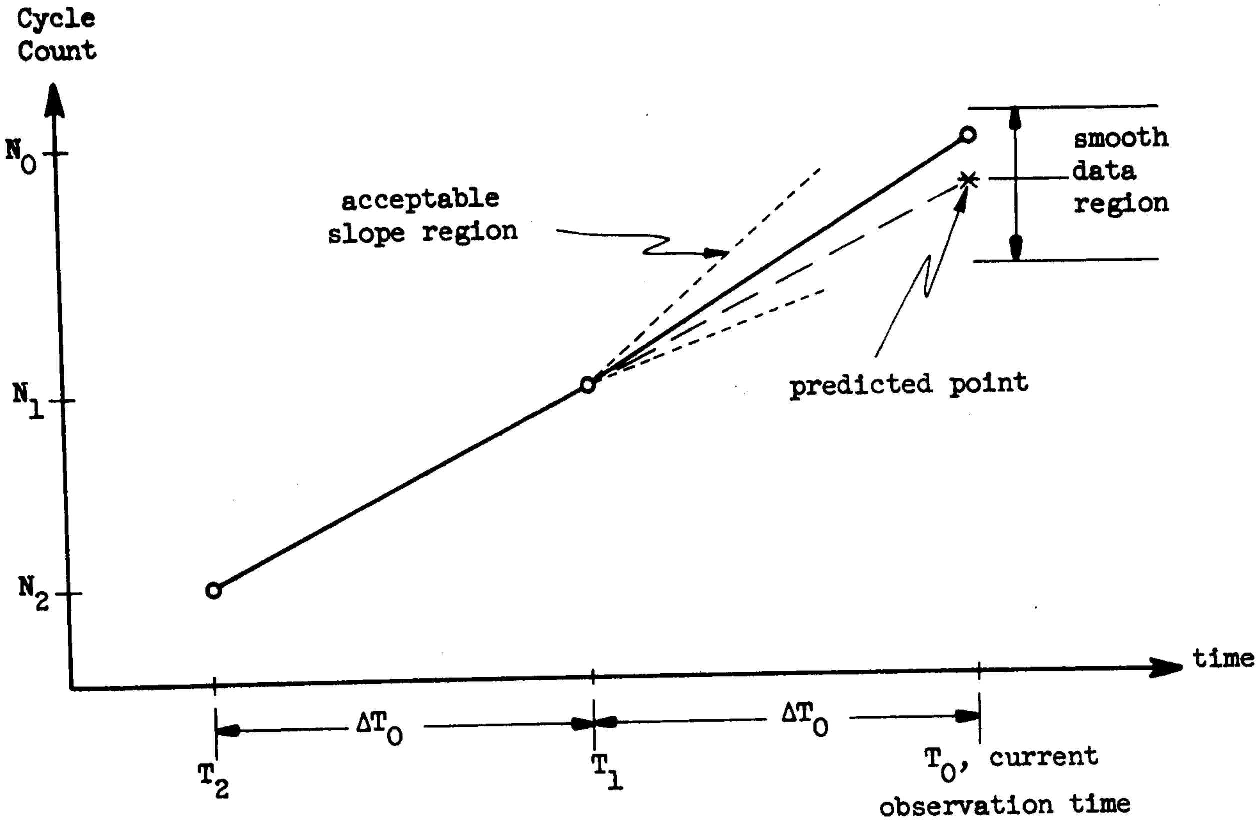


Figure 2.- Illustration of data editing.

Because the line-of-sight velocity of the LM with respect to the earth (for lunar ascent and descent) is always less than 6000 fps, note that

$$10^6 - 28{,}000 < (N_0 - N_1)/\Delta T_0 < 10^6 + 28{,}000 \text{ cycles/sec}$$

The predicted data point is given by the first-order Taylor series expansion

$$\hat{N}_0 = N_1 + \frac{N_1 - N_2}{T_1 - T_2}(T_0 - T_1)$$

The smoothness check is passed if

$$|N_0 - \hat{N}_0| < \Delta_{2MAX}$$

where a value will be derived for Δ_{2MAX} later.

Data editing is greatly simplified if there is an assurance of available data at every observation time; that is, no missing data. This requirement can be easily satisfied if, when a data point is missing, the current cycle count is assumed to be the same as the previous cycle count. This is perfectly safe to do because the gross reasonableness test would be failed in this case and the data point would be labeled bad. If there are data at every observation time, then

$$T_1 - T_2 = T_0 - T_1 = \Delta T_0,$$

and

$$\hat{N}_0 = 2N_1 - N_2$$

and

$$N_0 - \hat{N}_0 = N_0 - 2N_1 + N_2 = (N_0 - N_1) - (N_1 - N_2)$$

But the quantity on the right is the second difference of the cycle count, $\Delta_2 N_0$, which is approximately

$$\Delta_2 N_0 \approx 2 \frac{\omega_4 \nu_{tr}}{c} \Delta T_0^{\,2} \text{ (acceleration along the line of sight)}$$

Because the maximum acceleration of the LM is about 30 ft/sec^2, it follows that

$$|\Delta_2 N_0| < 140 \, \Delta T_0^{\,2} \text{ cycles}$$

because of LM acceleration. However, because there are random errors added to the measurements, it follows that

$$\sigma_{\Delta 2} = \sqrt{6}\ \sigma_N \qquad (\sigma_N \approx \frac{1}{3}\text{ cycle})$$

$$\approx .82 \text{ cycles}$$

Because the random error is primarily caused by a zero to 1 quantization error in the cycle counter, the maximum value of $\Delta_2 N_0$ caused by quantization error would be

$$|\Delta_2 N_0| < 2 \text{ cycles}$$

Thus, the value for Δ_{2MAX} may be chosen as

$$\Delta_{2MAX} \approx 2 + 140\ \Delta T_0^{\ 2} \text{ cycles}$$

In summary the current data is accepted if

$$\Delta_{1MIN} < N_0 - N_1 < \Delta_{1MAX}$$

and

$$|(N_0 - N_1) - (N_1 - N_2)| < \Delta_{2MAX}$$

where

$$\Delta_{1MIN} \approx (10^6 - 28{,}000)\ \Delta T_0 \text{ cycles}$$

$$\Delta_{1MAX} \approx (10^6 + 28{,}000)\ \Delta T_0 \text{ cycles}$$

$$\Delta_{2MAX} \approx 2 + 140\ \Delta T_0^{\ 2} \text{ cycles}$$

There is one further data editing check performed in another part of the program. Immediately before the state vector is corrected, the solution for the "rate bias", $b_L + \varepsilon_{\omega 3L}$, is checked to see if it is too large, based on the current measurements. If it is larger than some predetermined maximum value, then the residual for that station is set to zero, the corresponding row of the measurement matrix is zeroed, F(L+8) is set to zero, and the measurement weighting matrix, B, is re-estimated. The reason for this additional editing check is that with the normal editing a large rate bias error cannot be easily detected. Also, depending on the size of ΔT_0, a few wild data points can pass the normal editing.

The data status checks do the following

1. If data for station L is missing or bad for time T_0, the number in the station L cycle count cell is labeled bad by setting F(L+4) = 0.

2. Likewise, F(L+8) is set to zero (elsewhere in the program) if station L has too large a "rate bias" error, $b_L + \varepsilon_{\omega 3L}$.

3. F(L) is set to 1 if station L data goes from bad to good. When F(L) = 1, T_{OIL} and the station L integration constant, I_L, will be reinitialized. Also the corresponding row and column of the state error covariance matrix will be reinitialized.

4. F(L) is set to -1 if the station L location changes or if the transmitter location changes. When F(L) = -1, the I_L, T_{OIL}, and $\varepsilon_{\omega 3L}$ = X(L+13) are reinitialized, and the corresponding row and column of the state error covariance matrix will be reinitialized.

5. The F(L+12) flag is set to zero or 1 according to whether the station is a 3-way or 2-way station.

A block diagram of this logic is shown in figure 3. In this diagram

YY(L) is storage for the current cycle count from station L

$YY_p(L)$ is the past value of YY(L)

$$\Delta YY(L) = YY(L) - YY_p(L)$$

$\Delta YY_p(L)$ = past value of $\Delta YY(L)$

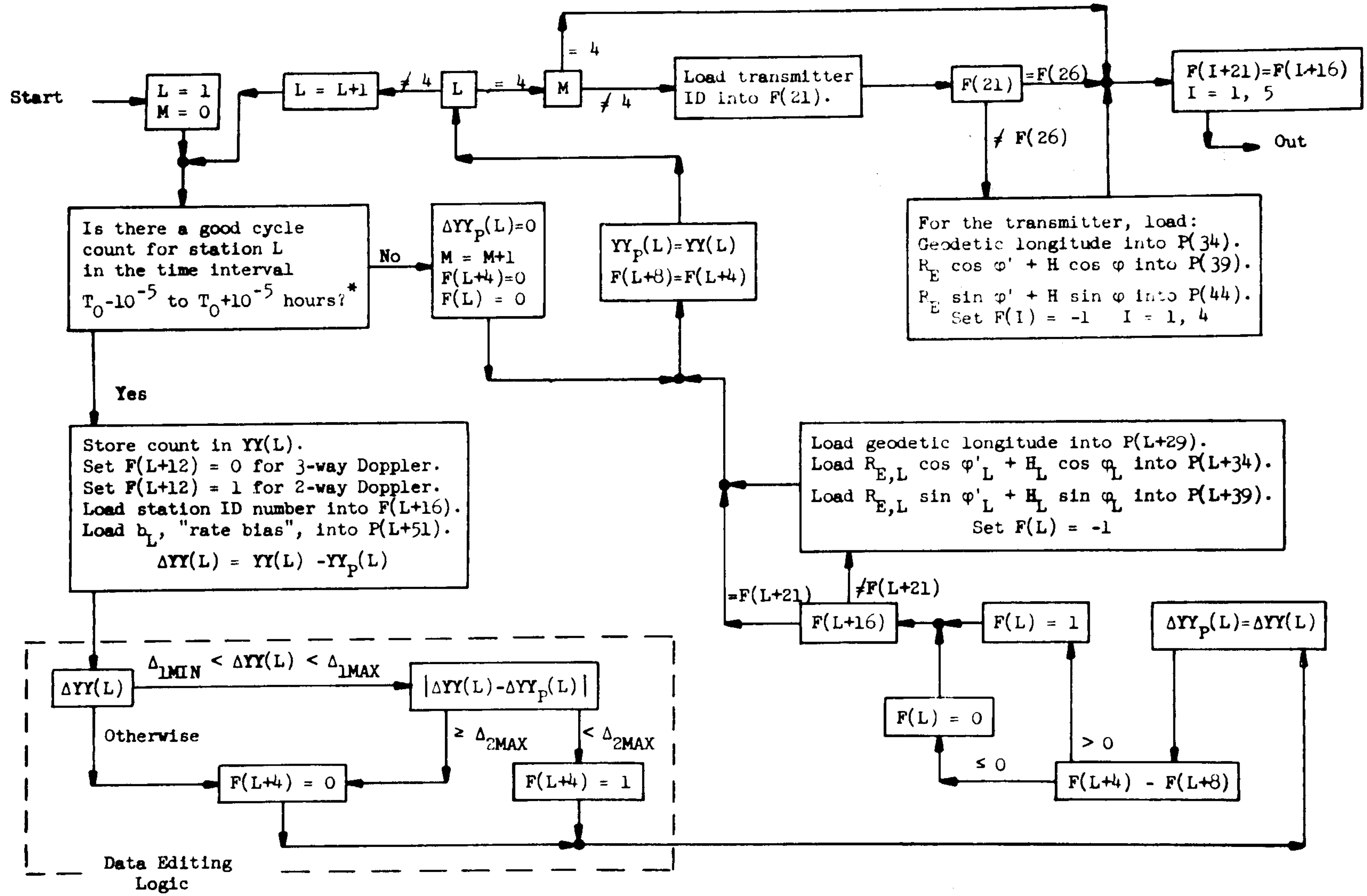


Figure 3.- Block diagram for data input logic and data status checks.

*The word good means that the tracking station attached a good label to the data.

10.0 INITIALIZATION OF MEASUREMENT CONSTANTS

When $F(L) = 1$, T_{OIL} and the station L integration constant, I_L, will be reinitialized. Also, the corresponding row and column of the state error covariance matrix will be reinitialized. In section 8 the initialization equation for I_L was given by

$$I_L = \frac{\omega_4 \nu_{tr}}{c} [|\underline{R}_V(T_{VL}) - \underline{R}_T(T_{TL})| + |\underline{R}_V(T_{VL}) - \underline{R}_{SL}(T_{OIL})|] - N^*_{FL}(T_{OIL})$$

When $F(L) = -1$, T_{OIL}, I_L and $\varepsilon_{\omega 3L} = X(L + 13)$ will be reinitialized; $X(L + 13)$ is initialized by setting it equal to zero. Likewise, the corresponding rows and columns of the state error covariance will be reinitialized. Whenever initialization takes place, the measurement is used in the initialization equations and should not be used again in the Kalman filter. To prevent this reuse, the appropriate row of the M matrix is zeroed out.

The block diagram for the initialization procedure is shown in figure 4. In this block diagram

$P(46) = \sigma^2_{\omega 3}$ for a 3-way station

$P(47) = \sigma^2_{\omega 3}$ for a 2-way station

$P(L+47) = \sigma^2_{\omega 3}$ for station L, L = 1, 2, 3, 4

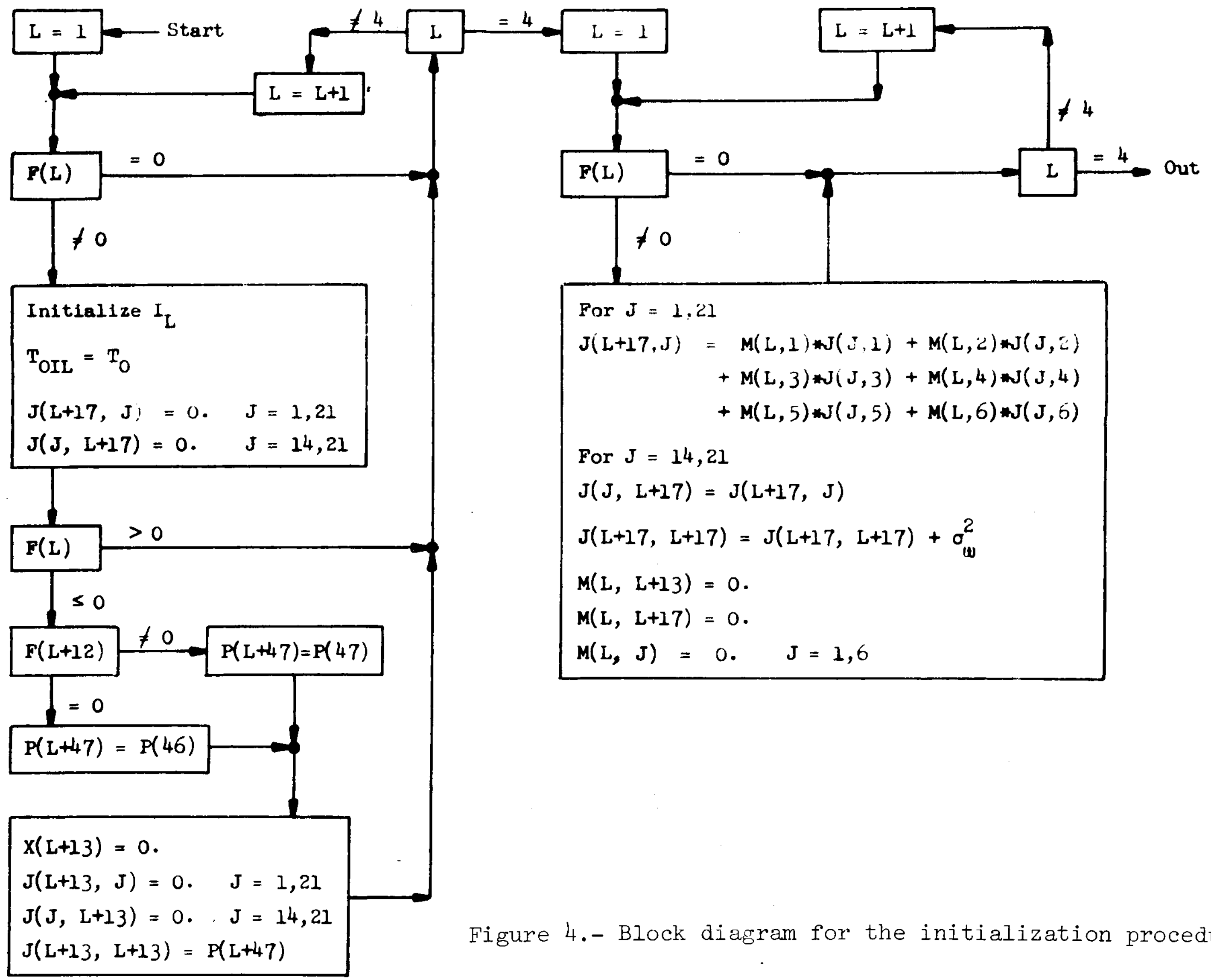


Figure 4.- Block diagram for the initialization procedure.

11.0 STATE AND COVARIANCE MATRIX UPDATE EQUATIONS

As in section 6, special procedures to save time are used to perform the following matrix operations: $D = JM^T$, $H = MD + W$, $H = H^{-1}$, $B = DH$, $J = J - BD^T$, and $\underline{X} = \underline{X} + B(\underline{Y}^* - \hat{\underline{Y}})$. The special procedures are given below and as part of the detailed program instructions.

In section 8, the M matrix was shown in the form

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{1,14} & 0 & 0 & 0 & m_{1,18} & 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{2,15} & 0 & 0 & 0 & m_{2,19} & 0 & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{3,16} & 0 & 0 & 0 & m_{3,20} & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{4,17} & 0 & 0 & 0 & m_{4,21} \end{bmatrix}$$

By the use of the special form of the M matrix, $D = J\,M^T$ is given by

DØ a I = 1,21

DØ a J = 1,4

```
a    D(I,J)  =  J(I,1)*M(J,1) + J(I,2)*M(J,2) + J(I,3)*M(J,3)
                + J(I,4)*M(J,4) + J(I,5)*M(J,5) + J(I,6)*M(J,6)
                + J(J+13, I)*M(J, J+13) + J(J+17, I)* M(J, J+17)
```

Note that J(I,J) I = 1,13 J = 14,21 is not used in the above algorithm.

H = MD + W is given by

```
     DØ b   I = 1,4
     DØ a   J = 1,I
a    H(I,J)  =  M(I,1)*D(1,J) + M(I,2)*D(2,J) + M(I,3)*D(3,J)
                + M(I,4)*D(4,J) + M(I,5)*D(5,J) + M(I,6)*D(6,J)
                + M(I, I+13)*D(I+13, J) + M(I, I+17)*D(I+17, J)
```

b   H(I,I) = H(I,I) + σ_w^2

Note that only the lower triangular part of the symmetric matrix H is calculated.

H = H^{-1} is given by

```
     H(1,1)  =  1.DO/H(1,1)
     DØ e   J = 1,3
     L = J+1
     DØ a   I = 1,J
     H(I,L)  =  0. DO
     DØ a   K = 1,J
a    H(I,L)  =  H(I,L) - H(L,K)*H(I,K)
     DØ b   K = 1,J
```

```
b     H(L,L)  =  H(L,L) + H(L,K)*H(K,L)
      H(L,L)  =  1. DO/H(L,L)
      DØ c   I = 1,J
c     H(L,I)  =  H(I,L)*H(L,L)
      DØ d   I = 1,J
      DØ d   K = I,J
      H(I,K)  =  H(I,K) + H(I,L)*H(L,K)
d     H(K,I)  =  H(I,K)
      DØ e   I = 1,J
e     H(I,L)  =  H(L,I)
```

Note that the above inversion algorithm is designed specifically to invert a symmetric matrix using only the lower triangular part of that matrix. The algorithm requires about half the number of multiplications and additions that a normal matrix multiplication requires. Also, no external storage is used.

B = DH is given by

```
      DØ a   I = 1,21
      DØ a   J = 1,4
a     B(I,J)  =  D(I,1)*H(1,J) + D(I,2)*H(2,J) + D(I,3)*H(3,J)
                 + D(I,4)*H(4,J)
```

J = J - B D^T is given by

```
      DØ a   I = 1,13
      DØ a   J = I,13
```

```
    J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                    -B(I,4)*D(J,4)
a   J(J,I)  =  J(I,J)
    DØ c    I = 14,21
    DØ b    J = 1,13
b   J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                    -B(I,4)*D(J,4)
    DØ c    J = I,21
    J(I,J)  =  J(I,J) -B(I,1)*D(J,1) -B(I,2)*D(J,2) -B(I,3)*D(J,3)
                    -B(I,4)*D(J,4)
c   J(J,I)  =  J(I,J)
```

Note that the above algorithm makes use of the symmetry of the J matrix, and, again, J(I,J) I = 1, 13 J = 14, 21 is not used or disturbed. This part of the J matrix has not been used anywhere in the previous sections. Thus, this block of 104 double precision words of nondestroyable storage is available for other uses within the program.

$\underline{x} = \underline{x} + B(\underline{y}^* - \hat{\underline{y}})$, where $\underline{y}^* - \hat{\underline{y}}$ is stored in Y(I), is given by

```
DØ a   I = 1,21
X(I)  =  X(I) + B(I,1)*Y(1) + B(I,2)*Y(2) + B(I,3)*Y(3)
              + B(I,4)*Y(4)
```

12.0 LIGHT TIME SUBROUTINE (SUBROUTINE TRANST)

This subroutine solves the transit time equations; that is, given a measurement observation time T_0, TRANST solves for T_V, the time at which the signal had to leave the vehicle in order to arrive at the receiving station at time T_0. TRANST also solves for T_T, the time at which the signal had to leave the transmitter in order to arrive at the vehicle at time T_V.

The subroutine estimates $\underline{R}_V(T_{VI}) - \underline{R}_{SI}(T_0)$, the range vector from the receiver to the vehicle, and $\underline{R}_V(T_{VI}) - \underline{R}_T(T_{TI})$, the range vector from the transmitter to the vehicle. TRANST also solves for the magnitude of these vectors.

The time for station I, T_{VI}, is obtained by iterating the equation

$$\begin{aligned} T_{VI} = T_0 - \frac{1}{c} \Big\{ & [x(T_F) + (T_{VI} - T_F)\, \dot{x}(T_F) - x_{SI}(T_0)]^2 \\ & + [y(T_F) + (T_{VI} - T_F)\, \dot{y}(T_F) - y_{SI}(T_0)]^2 \\ & + [z(T_F) + (T_{VI} - T_F)\, \dot{z}(T_F) - z_{SI}(T_0)]^2 \Big\}^{1/2} \end{aligned}$$

A value of $T_{VI} = T_F$ (T_F is the filter time tag of the state vector) is used to start the iteration. Because of the manner in which T_F is updated ($\Delta T_F = T_{V1} + \Delta T_0 - T_F$), $T_{VI} - T_F$ is always less than 0.02 second.

From the above equation, $T_{VI}^{(n)}$ may be written symbolically for the nth iterated solution as

$$T_{VI}^{(n)} = f[T_{VI}^{(n-1)}]$$

The error, e, in the solution for T_{VI}, after n iterations, is given by the equation

$$|e^{(n)}| = |e^{(0)}| M^n$$

where M is the maximum absolute value of df/dT_{VI} in the vicinity of T_{VI}. The equation for df/dT_{VI} is

$$\frac{df}{dT_{VI}} = -\frac{1}{c}\frac{\underline{R}_{V/S} \cdot \dot{\underline{R}}_{V/E}}{|\underline{R}_{V/S}|}$$

where V/S refers to vehicle with respect to the station, and V/E refers to vehicle with respect to the earth. For a maximum vehicle velocity of 10^4 fps, M can be evaluated as

$$M \leq \frac{10^4}{10^9} = 10^{-5}$$

Because $|e^{(0)}|$ is less than $2 \cdot 10^{-2}$ seconds

$$|e^{(n)}| \leq 2 \cdot 10^{-2} (10^{-5})^n$$

for which a value of n = 1 will be used. Thus

$$|e^{(1)}| \leq 2 \cdot 10^{-7} \text{ seconds}$$

After T_{VI} is determined, the position of the vehicle can be calculated in the following manner.

$$x(T_{VI}) = x(T_F) + (T_{VI} - T_F)\, \dot{x}(T_F)$$

$$y(T_{VI}) = y(T_F) + (T_{VI} - T_F)\, \dot{y}(T_F)$$

$$z(T_{VI}) = z(T_F) + (T_{VI} - T_F)\, \dot{z}(T_F)$$

Also, the vector components $x(T_{VI}) - x_{SI}(T_0)$, $y(T_{VI}) - y_{SI}(T_0)$, $z(T_{VI}) - z_{SI}(T_0)$ and the absolute magnitude of this vector will be calculated.

In the calculation of T_{VI}, since the state vector is in MNBY coordinates, the four station location vectors must also be in MNBY coordinates. In true-of-date coordinates, the station location is given by

$$x'_{SI}(T_0) = P(I+34)*\cos[P(I+29) + \omega_{Earth}\, T_0]$$

$$y'_{SI}(T_0) = P(I+34)*\sin[P(I+29) + \omega_{Earth}\, T_0]$$

$$z'_{SI}(T_0) = P(I+39)$$

where P(I+29), P(I+34), and P(I+39) are obtained from the station characteristics table (section 9, fig. 2). These locations are then transformed to MNBY coordinates by the RNP matrix. This requires four coordinate transformations (one for each of the four stations).

$$\begin{bmatrix} x_{SI}(T_0) \\ y_{SI}(T_0) \\ z_{SI}(T_0) \end{bmatrix} = (RNP)^T \begin{bmatrix} x'_{SI}(T_0) \\ y'_{SI}(T_0) \\ z'_{SI}(T_0) \end{bmatrix}$$

The time T_{TI} is obtained from

$$T_{TI} = T_{VI} - \frac{1}{c}\left[[x(T_{VI}) - x_T(\hat{T}_T)]^2 + [y(T_{VI}) - y_T(\hat{T}_T)]^2 + [z(T_{VI}) - z_T(\hat{T}_T)]^2 \right]^{1/2}$$

where

$$\hat{T}_T = T_{V1} - (T_0 - T_{V1})$$

and where

$$\begin{bmatrix} x_T(\hat{T}_T) \\ y_T(\hat{T}_T) \\ z_T(\hat{T}_T) \end{bmatrix} = (RNP)^T \begin{bmatrix} P(39)*\cos[P(34) + \omega_E \hat{T}_T] \\ P(39)*\sin[P(34) + \omega_E \hat{T}_T] \\ P(44) \end{bmatrix}$$

Note that

$$\begin{bmatrix} \dot{x}_T(\hat{T}_T) \\ \dot{y}_T(\hat{T}_T) \\ \dot{z}_T(\hat{T}_T) \end{bmatrix} = (RNP)^T \begin{bmatrix} -\omega_E \; P(39) \; \sin[P(34) + \omega_E \hat{T}_T] \\ \omega_E \; P(39) \; \cos[P(34) + \omega_E \hat{T}_T] \\ 0 \end{bmatrix}$$

Thus, $x_T(T_{TI})$, $y_T(T_{TI})$, and $z_T(T_{TI})$ are given by

$$\begin{bmatrix} x_T(T_{TI}) \\ y_T(T_{TI}) \\ z_T(T_{TI}) \end{bmatrix} = \begin{bmatrix} x_T(\hat{T}_T) \\ y_T(\hat{T}_T) \\ z_T(\hat{T}_T) \end{bmatrix} + (T_{TI} - \hat{T}_T) \begin{bmatrix} \dot{x}_T(\hat{T}_T) \\ \dot{y}_T(\hat{T}_T) \\ \dot{z}_T(\hat{T}_T) \end{bmatrix}$$

The vector components $x(T_{VI}) - x_T(T_{TI})$, $y(T_{VI}) - y_T(T_{TI})$, $z(T_{VI}) - z_T(T_{TI})$ and the absolute magnitude of this vector can now be calculated.

This method for solving the transit time equations requires only six multiplications by the RNP matrix and only five calls to the sine and cosine subroutines. A more straightforward approach would have resulted in 12 RNP multiplications and 12 calls to the sine and cosine subroutines.

Despite the fact that only a single iteration is used to obtain T_{VI} and T_{TI}, the accuracy of the solutions is more than adequate. Based on a maximum vehicle velocity of 10^4 fps with respect to the earth, the maximum error in T_{VI} is $2 \cdot 10^{-7}$ seconds. The maximum error in calculating $\underline{R}_V(T_{VI}) - \underline{R}_{SI}(T_O)$ is 0.002 ft. The maximum error in calculating T_{TI} is about $0.4 \cdot 10^{-7}$ second.

13.0 SUPERVISORY LOGIC

The overall block diagram of the program is shown in figure 5. The program has four main areas of specialties. Blocks 1 and 2 are first-time-only blocks, used to initialize various constants and variables. Some of these quantities are reset in the restart block in the event that restart conditions occur. Blocks 3 through 6 are concerned with the dynamics. Blocks 7 through 14 process the measurements. Blocks 15 and 16 integrate the current best estimates of position and velocity ahead, or back, to a desired time, T_D. The output position and velocity in block 16 will be in MNBY selenocentric coordinates with units of earth radii and earth radii per hour. The time tag will be in hours from midnight of the launch day.

The subroutine TRAJ, mentioned in the overall block diagram, integrates the first six elements of the state vector over the interval ΔT seconds, updates time, and integrates the moon's position and velocity over the interval ΔT seconds. Subroutine TRAJ has its maximum integration step size limited to ΔT_{MAX}. This is to prevent an excessive truncation error in the state estimate if an unusually large ΔT is called for. A block diagram that illustrates how ΔT is limited is shown in figure 6.

New variables used in the block diagram are defined below.

T_F = filter time tag of the state vector

T_{RL} = real (current) ground time, hours from midnight of launch day

T_{V1} = time at which a signal would have to leave the spacecraft in order to arrive at the first ground tracking station at time T_0, the observation time

$(T_F+P_{119})R$ = T_F+P_{119} rounded to the nearest 2/36 000 hour (nearest 2/10 second) or nearest 4/36 000 hour, depending upon whether data is processed every 0.2 second or every 0.4 second

P_{119} = approximate value of $T_0 - T_{V1}$ (P_{119} is an input constant)

T_0 = observation time

T_L = constant lag time between observations and T_{RL}

ΔT_0 = time interval between observations

ΔT_F	=	integration step size for the filter
ΔT	=	integration step size used in TRAJ
ΔT_{MAX}	=	maximum allowable value of ΔT
T_D	=	desired time of output state vector
T_{DL}	=	last cycle's value of T_D

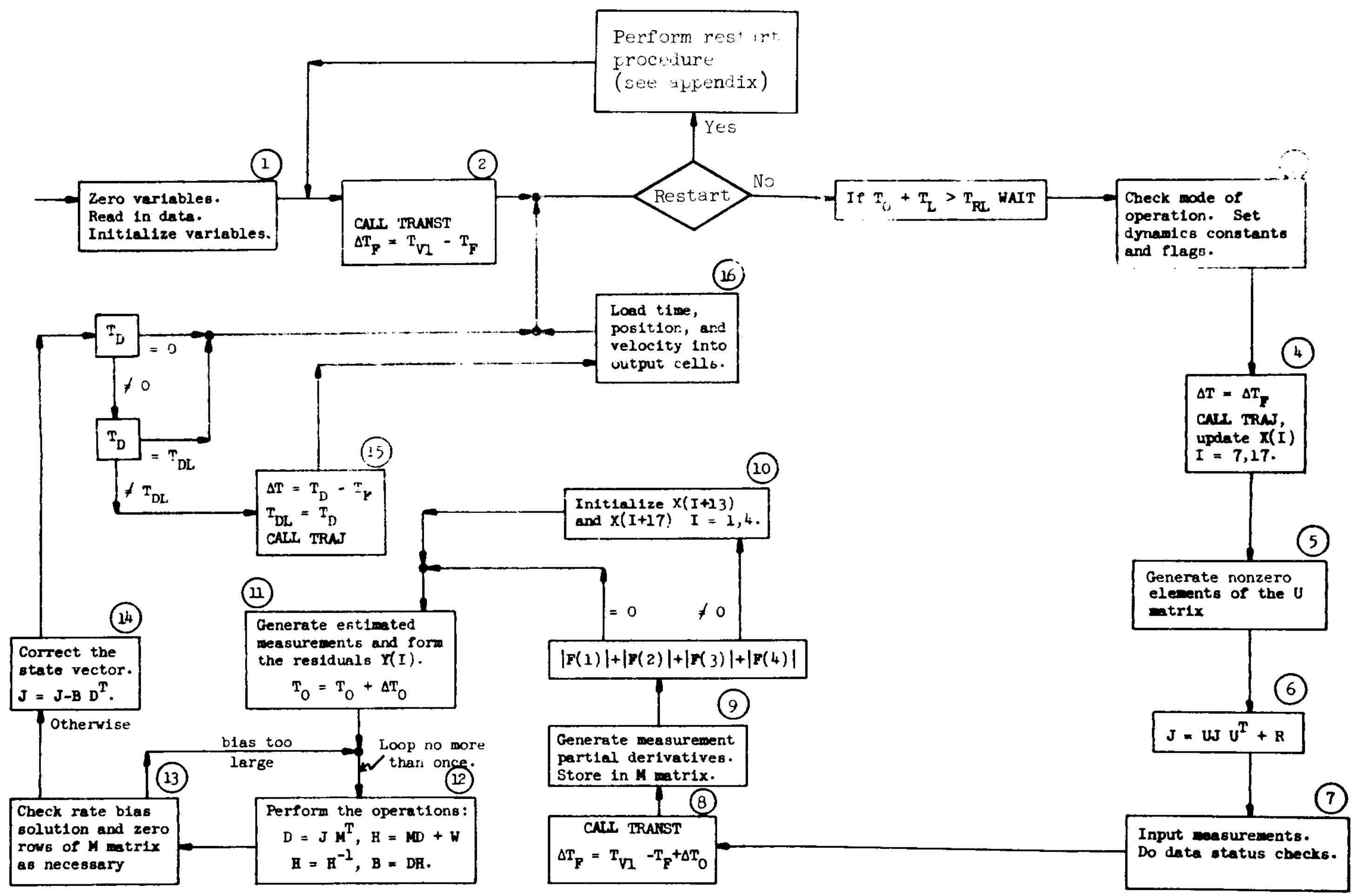


Figure 5.- Overall block diagram.

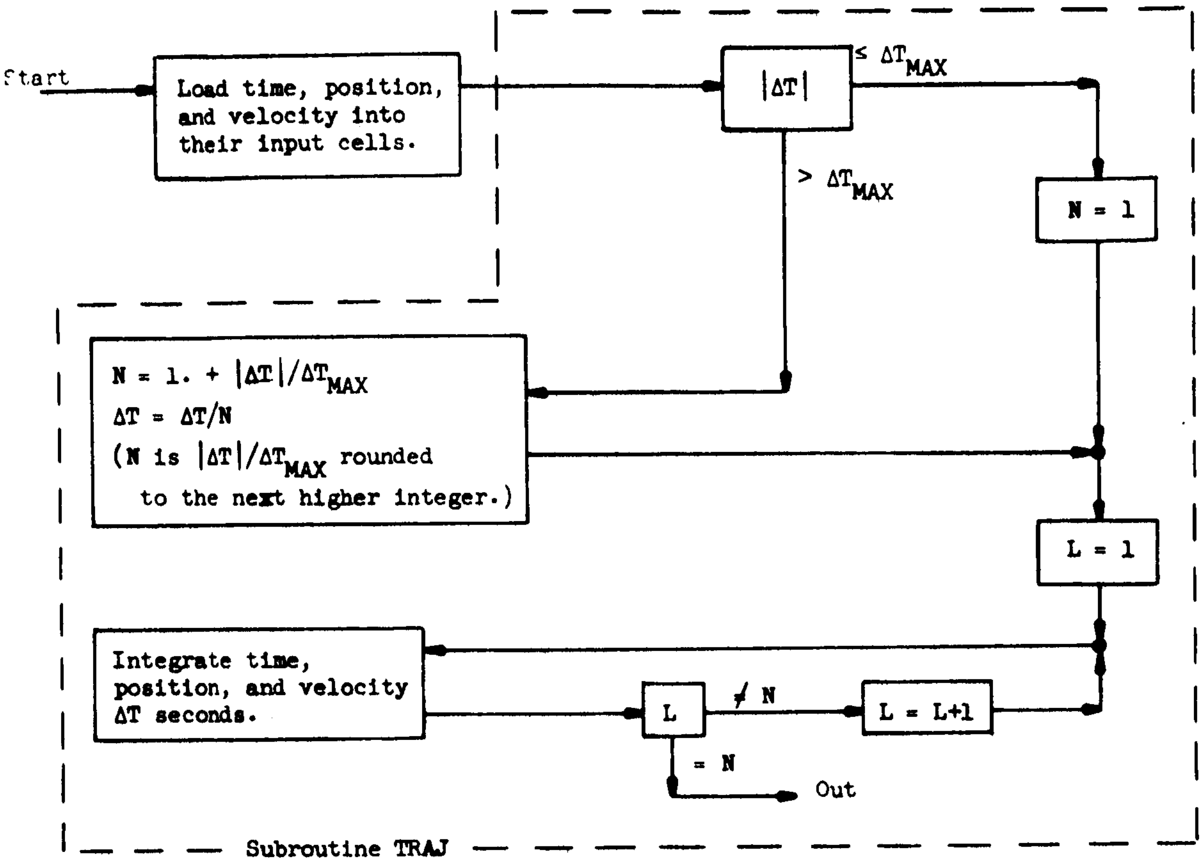


Figure 6.- Logic for controlling integration step size in subroutine TRAJ.

14.0 DETAILED PROGRAMING INSTRUCTIONS

The previous sections of this report have outlined the engineering equations and logic and were intended only to supply general background information. The following sections present the actual detailed programing requirements.

Ordinarily, a 21-state variable filter for use in real-time is indeed a formidable proposition since, for example, it would be nearly 13 times slower in execution time than a 9-state variable filter.

To make the 21-state variable filter practical for use in real time, the formulation of the engineering equations has been done by considering the number of numerical operations performed by the digital computer.

The detailed programing instructions will be presented in notation similar to FORTRAN. Namely, DØ loops will be used, not because of any expertise in programing, but because they offer the engineer an elegantly simple way to write large groups of equations. Also, the notation used will be understandable by the engineer; while, at the same time, it will be directly translatable into FORTRAN or machine language by the programer. Presentation of the programing instructions in this clearly understandable form makes it possible for the computer program to be implemented quickly with the least possible chance of error.

Variables used within the program may be categorized broadly into three types

1. Fixed point variables, used primarily as logic flags

2. Double-precision, floating-point, permanent-storage variables, used wherever quantities must be saved from one cycle to the next

3. Double-precision, floating-point, temporary-storage variables, used for quantities that need not be saved from one cycle to the next

The next three subsections will discuss these three types of variables in more detail. The definitions of these variables serve as a bridge between the engineering notation and the computer program notation.

14.1 The Fixed-Point Variables

There will be 48 fixed-point variables

$$F(L) \quad L = 1, 2, \ldots, 48$$

The first 35 of these variables will be defined here. The last 13 will be for future program use if the need should arise.

F(L) L = 1, 4

F(L) = 0, do not reset I_L, T_{OIL}, and $\epsilon_{\omega 3L}$

F(L) = 1, reset I_L and T_{OIL} (caused by bad data)

F(L) = -1, reset I_L, T_{OIL}, and $\epsilon_{\omega 3L}$ (caused by station location change)

F(L+4) L = 1, 4

F(L+4) = 0, current data bad or missing for station L

F(L+4) = 1, current data good for station L

F(L+8) L = 1, 4

F(L+8) = last value of F(L+4)

(Note: it may be desirable to display F(L+8).)

F(L+12) L = 1, 4

F(L+12) = 0 for 3-way data from station L

F(L+12) = 1 for 2-way data from station L

F(L+16) L = 1, 4 and F(21)

F(L+16) is the station L identification number; F(21) is the transmitter identification number

F(L+21) L = 1, 5

F(L+21) = the last value of F(L+16)

F(31)

F(31) = 0 for no good tracking data

F(31) = 1 if there is good tracking data

F(32)

F(32) = 0 for descent

F(32) = 1 for ascent from surface of the moon

F(33)

F(33) is the previous value of F(31)

F(34)

F(34) = 0 for free-flight before powered descent

F(34) = 1 otherwise

F(35)

F(35) = 0 under normal program operating conditions

F(35) = 1 under program restart conditions

14.2 Permanent Storage Variables

The following variables are double-precision, floating-point variables whose values must be preserved from one cycle to the next.

X(I) I = 1, 21

X(I) is the state vector, the elements of which were described in section 4

YY(I) I = 1, 4

The actual measurement vector, N^*_{FI}, described in section 8

JX(I, J) I = 1, 21 J = 1, 21

J(I, J) is the symmetric, state error-covariance matrix defined in section 3; J(I, J) I = 1, 13 J = 14, 21 is not used by the program. To avoid conflict with the integer J, the J(I, J) matrix will be designated as JX(I, J) in the program.

P(I)   I = 1, 180

P(I) consists of input constants and program generated constants. The first 126 P's are defined below. The remaining 54 P's are reserved for future use by the program. If storage is a problem, the first 104 P's may be equivalenced to JX(I, J) I = 1, 13 J = 14, 21.

$$\begin{bmatrix} P(1) & P(2) & P(3) \\ P(4) & P(5) & P(6) \\ P(7) & P(8) & P(9) \end{bmatrix}$$ = the RNP matrix. $(RNP)^T$ is used to convert station location to MNBY coordinates.

P(10) = $\exp[-|\Delta T_0|/T_{\omega P}]$

P(11) = $\exp[-|\Delta T_0|/T_{\omega Y}]$

P(12) = $\exp[-|\Delta T_0|/T_{\epsilon \hat{M}}]$

P(13) = $\exp[-|\Delta T_0|/T_{\epsilon I}]$

P(14) = $\exp[-|\Delta T_0|/T_{\omega 3}]$

P(15) = T_F, time tag of state vector in the filter

P(16) = ΔT_F, filter integration step size

P(17) = ΔT_{MAX}, the maximum ΔT used by TRAJ

P(18) = T_0, current observation time

P(19) = ΔT_0, time interval between usable observations

P(20) = T_L, constant time lag between real time and T_0

P(21) = T_D, desired time of output state vector

P(22) = T_{DL}, the last value of T_D

P(23) = time tag for output from program

$$\left.\begin{array}{l} P(24) = x_{V/M} \\ P(25) = y_{V/M} \\ P(26) = z_{V/M} \\ P(27) = \dot{x}_{V/M} \\ P(28) = \dot{y}_{V/M} \\ P(29) = \dot{z}_{V/M} \end{array}\right\}$$

LM position and velocity output from the program, with time tag of P(23), and in MNBY selenocentric coordinates

P(L+29) = longitude (+ east) of station L, L = 1, 4

P(34) = longitude of transmitter

P(L+34) = $R_{EL} \cos \varphi'_L + H_L \cos \varphi_L = \sqrt{(x'_{SL})^2 + (y'_{SL})^2}$, L = 1, 4

P(39) = $R_E \cos \varphi' + H \cos \varphi$ for transmitter

P(L+39) = $R_{EL} \sin \varphi'_L + H_L \sin \varphi_L = z'_{SL}$, L = 1, 4

P(44) = $R_E \sin \varphi' + H \sin \varphi$ for transmitter

P(45) = maximum allowable value of $b_L + \epsilon_{\omega 3L}$ before station L's data is deleted by the filter

P(46) = $\sigma^2_{\omega 3}$ for a 3-way station

P(47) = $\sigma^2_{\omega 3}$ for a 2-way station

P(L+47) = $\sigma^2_{\omega 3}$ for station L, L = 1, 4

P(L+51) = b_L, the a priori estimate of the station L "rate bias" error, L = 1, 4

P(L+55) = value for last cycle of N^*_{FL} = YY(L), L = 1, 4

$$\left.\begin{aligned} P(60) &= x_{M/E} \\ P(61) &= y_{M/E} \\ P(62) &= z_{M/E} \\ P(63) &= \dot{x}_{M/E} \\ P(64) &= \dot{y}_{M/E} \\ P(65) &= \dot{z}_{M/E} \end{aligned}\right\}$$

position and velocity of the moon with respect to the earth in geocentric, MNBY coordinates

P(66) = μ_E, earth's gravitational constant

P(67) = μ_M, moon's gravitational constant

P(68) = $\mu_E + \mu_M$

P(69) = g in $gI\dot{M}/M$

P(70) = ω_{PN}, nominal value of $\dot{\psi}_P$

P(71) = ω_{YN}, nominal value of $\dot{\psi}_Y$

P(72) = $\dot{M}_N (\dot{M}_N \leq 0)$, nominal value of $\dot{M}$

P(73) = I_N, nominal value of specific impulse

$$\begin{bmatrix} P(74) & P(77) & P(80) \\ P(75) & P(78) & P(81) \\ P(76) & P(79) & P(82) \end{bmatrix} = \begin{bmatrix} c_1 & c_4 & c_7 \\ c_2 & c_5 & c_8 \\ c_3 & c_6 & c_9 \end{bmatrix}$$

$P(83) = 1 - [\exp(-|\Delta T_0|/T_{\omega P})]^2$

$P(84) = 1 - [\exp(-|\Delta T_0|/T_{\omega Y})]^2$

$P(85) = 1 - [\exp(-|\Delta T_0|/T_{\epsilon \dot{M}})]^2$

$P(86) = 1 - [\exp(-|\Delta T_0|/T_{\epsilon I})]^2$

$P(87) = 1 - [\exp(-|\Delta T_0|/T_{\omega 3})]^2$

$P(88) = \sigma^2_{\omega P}$

$P(89) = \sigma^2_{\omega Y}$

$P(90) = \sigma^2_{\epsilon \dot{M}}$

$P(91) = \sigma^2_{\epsilon I}$

$P(92) = \sigma^2_{ag}\ \Delta T_0{}^2$, velocity variance due to random gravitational acceleration

$P(93) = \dfrac{\omega_4\ \nu_{tr}}{c}$, a measurement data constant

$P(94) = \omega_3$, biasing frequency for all stations

$P(95) = c$, the speed of light

$P(96) = \omega_{Earth}$, angular velocity of the earth

$P(L+96) = T_{OIL}$ for station L, L = 1, 4

$P(101) = \sigma^2_w$, variance of measurement noise

$P(102)$ = initial value of $J(I, I)$ I = 1, 3 for descent

$P(103)$ = initial value of $J(I, I)$ I = 4, 6 for descent

P(104) = initial value of J(I, I) I = 7, 8 for descent.

P(105) = initial value of J(I, I) I = 9 for descent

P(106) = initial value of J(I, I) I = 1, 3 for ascent

P(107) = initial value of J(I, I) I = 4, 6 for ascent

P(108) = initial value of J(I, I) I = 7, 8 for ascent

P(109) = initial value of J(I, I) I = 9 for ascent

P(L+109) = $\Delta YY_p(L)$ L = 1, 4, the past value of $\Delta YY(L)$

P(114) = Δ_{1MIN}, the minimum allowable value of $\Delta YY(L)$

P(115) = Δ_{1MAX}, the maximum allowable value of $\Delta YY(L)$

P(116) = Δ_{2MAX}, the maximum allowable value of $|\Delta YY(L) - \Delta YY_p(L)|$

P(117) = initial value of mass for descent

P(118) = initial value of mass for ascent

P(119) = approximate value of the transit time of electromagnetic waves from the spacecraft to the receiving station

P(120) = motor on time, in hours from midnight of launch day, for descent; motor on time should be at the start of ullage

P(121) = initial value of $\dot{M}_N$ for ascent

P(122) = powered flight value of $\sigma^2_{\epsilon\dot{M}}$, used when tracking data is available

P(123) = powered flight value of $\sigma^2_{\varepsilon \dot{M}}$, used when no tracking data is available

P(124) = time interval for processing free flight tracking data prior to powered descent initiation

P(125) = time offset to engine-on-time for the ascent phase

P(126) = time offset to engine-on-time for the descent phase

14.3 Temporary Storage Variables

All temporary storage variables are double precision. The sizes of the various matrices and vectors are shown here.

M(4, 21)	D(21, 4)	H(4, 4)	B(21, 4)
A(13, 13)	Y(4)	E(169)	

To save storage, the vector E(I) should be equivalenced to the 169 elements of A(I, J). Also, the matrix D(I, J) should be equivalenced to the first 84 elements of A(I, J), and B(I, J) should be equivalenced to the next 84 elements of A(I, J). However, note that the operation of the program is not dependent upon the equivalence of the above variables. The equivalencing is only a storage saving feature.

The total storage required by the variables mentioned in this section, and the preceding two sections, is 1782 words of core storage. Note that a single precision version of this program (practical only on a CDC machine) requires only 915 words of core storage.

14.4 The Subroutine TRANST Programing Instructions

In section 12, the engineering equations for the subroutine TRANST were discussed. The engineering equations are "linked" to the programing instructions by the definitions of the E cells shown here.

$E(1) = \omega_{Earth} \times T_0$

<u>For L = 1, 2, 3, 4</u>

$E(L+1) = x'_{SL}(T_0)$ $\quad E(L+5) = y'_{SL}(T_0)$

$E(L+9) = x_{SL}(T_0)$ $\quad E(L+13) = y_{SL}(T_0)$ $\quad E(L+17) = z_{SL}(T_0)$

$E(L+21) = T_{VL}$

$E(L+25) = x(T_{VL})$ $\quad E(L+29) = y(T_{VL})$ $\quad E(L+33) = z(T_{VL})$

$E(L+37) = x(T_{VL}) - x_{SL}(T_0)$

$E(L+41) = y(T_{VL}) - y_{SL}(T_0)$

$E(L+45) = z(T_{VL}) - z_{SL}(T_0)$

$E(L+49) = |\underline{R}_{V/E}(T_{VL}) - \underline{R}_{SL}(T_0)|$

$E(54) = \hat{T}_T$ $\quad E(55) = P(34) + \omega_E \times \hat{T}_T$

$E(56) = x'_T(\hat{T}_T)$ $\quad E(57) = y'_T(\hat{T}_T)$

$E(58) = \dot{x}'_T(\hat{T}_T)$ $\quad E(59) = \dot{y}'_T(\hat{T}_T)$

$E(60) = x_T(\hat{T}_T)$ $\quad E(61) = y_T(\hat{T}_T)$ $\quad E(62) = z_T(\hat{T}_T)$

$E(63) = \dot{x}_T(\hat{T}_T)$ $\quad E(64) = \dot{y}_T(\hat{T}_T)$ $\quad E(65) = \dot{z}_T(\hat{T}_T)$

<u>For L = 1, 2, 3 4</u>

$E(L+65) = T_{TL}$

$E(L+69) = x(T_{VL}) - x_T(T_{TL})$

$E(L+73) = y(T_{VL}) - y_T(T_{TL})$

$E(L+77) = z(T_{VL}) - z_T(T_{TL})$

$$E(L+81) = |\underline{R}_{V/E}(T_{VL}) - \underline{R}_T(T_{TL})|$$

$$E(L+85) = T_{VL} - T_F$$

$$E(L+89) = T_{TL} - \hat{T}_T$$

The detailed programing instructions are as follows

```
      E(1)  =  P(96)*P(18)
      DØ a  L = 1, 4
      E(L+1)  =  P(L+34)*DCØS(P(L+29) + E(1))
      E(L+5)  =  P(L+34)*DSIN(P(L+29) + E(1))
      E(L+9)  =  P(1)*E(L+1) + P(4)*E(L+5) + P(7)*P(L+39)
      E(L+13)  =  P(2)*E(L+1) + P(5)*E(L+5) + P(8)*P(L+39)
      E(L+17)  =  P(3)*E(L+1) + P(6)*E(L+5) + P(9)*P(L+39)
      E(L+21)  =  P(18) - (DSQRT((X(1) - E(L+9))**2
                + (X(2) - E(L+13))**2 + (X(3) - E(L+17))**2))/P(95)
      E(L+85)  =  E(L+21) - P(15)
      E(L+25)  =  X(1) + E(L+85)*X(4)
      E(L+29)  =  X(2) + E(L+85)*X(5)
      E(L+33)  =  X(3) + E(L+85)*X(6)
      E(L+37)  =  E(L+25) - E(L+9)

      E(L+41)  =  E(L+29) - E(L+13)
      E(L+45)  =  E(L+33) - E(L+17)
a     E(L+49)  =  DSQRT(E(L+37)**2 + E(L+41)**2 + E(L+45)**2)
      E(54)  =  E(22) - P(18) + E(22)
      E(55)  =  P(34) + P(96)*E(54)
```

```
    E(56)   =  P(39)*DCØS(E(55))
    E(57)   =  P(39)*DSIN(E(55))
    E(58)   =  -P(96)*E(57)
    E(59)   =  P(96)*E(56)
    DØ b   I = 1, 3
    E(I+59)  =  P(I)*E(56) + P(I+3)*E(57) + P(I+6)*P(44)
b   E(I+62)  =  P(I)*E(58) + P(I+3)*E(59)
    DØ c   L = 1, 4
    E(L+65)  =  E(L+21) - (DSQRT((E(L+25) - E(60))**2
              + (E(L+29) - E(61))**2 + (E(L+33) - E(62))**2))/P(95)
    E(L+89)  =  E(L+65) - E(54)
    E(L+69)  =  E(L+25) - E(60) - E(L+89)*E(63)
    E(L+73)  =  E(L+29) - E(61) - E(L+89)*E(64)
    E(L+77)  =  E(L+33) - E(62) - E(L+89)*E(65)
c   E(L+81)  =  DSQRT(E(L+69)**2 + E(L+73)**2 + E(L+77)**2)
```

Note that X(I), P(I), and E(I) should be in common.

14.5 The Subroutine TRAJ Programing Instructions

This subroutine will integrate the first six elements of the state vector, and the position and velocity of the moon over the interval ΔT seconds according to the equations outlined in section 5. The maximum step size is controlled by the logic shown in figure 6. At the present time, it is not anticipated that this option will ever be exercised. It is included only as a safety precaution to avoid an overly large integration step. This option is designed primarily for free-flight integration. If the option is exercised in powered flight, TRAJ assumes that ψ_P, ψ_Y and M are constant over the total integration step.

As shown here, the definitions of the E cells link the engineering equations to the programing instructions.

E(1) = time of the state estimate

$E(2) = x$ $E(3) = y$ $E(4) = z$

$E(5) = \dot{x}$ $E(6) = \dot{y}$ $E(7) = \dot{z}$

$E(8) = x_{M/E}$ $E(9) = y_{M/E}$ $E(10) = z_{M/E}$

$E(11) = \dot{x}_{M/E}$ $E(12) = \dot{y}_{M/E}$ $E(13) = \dot{z}_{M/E}$

$E(14) = x - x_{M/E}$ $E(15) = y - y_{M/E}$ $E(16) = z - z_{M/E}$

$E(17) = \omega_{PN} + \epsilon_{\omega P}$ $E(18) = \omega_{YN} + \epsilon_{\omega Y}$ $E(19) = \dot{M}_N + \epsilon_{\dot{M}}$

$E(20) = \sin \psi_P$ $E(21) = \sin \psi_Y$

$E(22) = \cos \psi_P$ $E(23) = \cos \psi_Y$

$E(24) = \sin \psi_Y \sin \psi_P$ $E(25) = \sin \psi_Y \cos \psi_P$

$E(26) = \cos \psi_Y \sin \psi_P$ $E(27) = \cos \psi_Y \cos \psi_P$

$E(28) = |\underline{R}_{V/E}|^2$ $E(29) = |\underline{R}_{V/E}|$ $E(30) = \mu_E/|\underline{R}_{V/E}|^3$

$E(31) = \Delta T$ $E(32) = \Delta T^2/2$ $E(33) = \Delta T^3/6$

$E(34) = |\underline{R}_{M/E}|^2$ $E(35) = |\underline{R}_{M/E}|$ $E(36) = \mu_M/|\underline{R}_{M/E}|^3$

$E(37) = \frac{g}{M}$ $E(38) = \frac{g}{M}(I_N + \epsilon_I)$ $E(39) = \frac{g}{M}(\dot{M}_N + \epsilon_{\dot{M}})$

$E(40) = |\underline{R}_{V/E} - \underline{R}_{M/E}|^2$ $E(41) = |\underline{R}_{V/E} - \underline{R}_{M/E}|$ $E(42) = \mu_M/|\underline{R}_{V/E} - \underline{R}_{M/E}|^3$

$$E(43) = \frac{x}{|\underline{R}_{V/E}|} \qquad E(44) = \frac{y}{|\underline{R}_{V/E}|} \qquad E(45) = \frac{z}{|\underline{R}_{V/E}|}$$

$$E(46) = C_x \qquad E(47) = C_y \qquad E(48) = C_z$$

$$E(49) = \frac{x_{M/E}}{|\underline{R}_{M/E}|} \qquad E(50) = \frac{y_{M/E}}{|\underline{R}_{M/E}|} \qquad E(51) = \frac{z_{M/E}}{|\underline{R}_{M/E}|}$$

$$E(52) = \frac{\mu_E+\mu_M}{|\underline{R}_{M/E}|^3} \qquad E(53) = \frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} + \frac{\mu_E}{|\underline{R}_{V/E}|^3}$$

$$E(54) = \frac{-\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} + \frac{\mu_M}{|\underline{R}_{M/E}|^3}$$

$$E(55) = \frac{x - x_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|} \qquad E(56) = \frac{y - y_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|} \qquad E(57) = \frac{z - z_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$$

$$E(58) = \frac{g}{M}(I_N+\epsilon_I)(\dot{M}_N+\epsilon_{\dot{M}}) \qquad E(59) = \frac{g}{M}(I_N+\epsilon_I)(\dot{M}_N+\epsilon_{\dot{M}})\frac{1}{M}$$

$$E(60) = \frac{\partial\ddot{x}}{\partial x} \qquad E(61) = \frac{\partial\ddot{x}}{\partial y} \qquad E(62) = \frac{\partial\ddot{x}}{\partial z}$$

$$E(63) = \frac{\partial\ddot{x}}{\partial x_{M/E}} \qquad E(64) = \frac{\partial\ddot{x}}{\partial y_{M/E}} \qquad E(65) = \frac{\partial\ddot{x}}{\partial z_{M/E}}$$

$$E(66) = \frac{\partial\ddot{x}}{\partial\psi_P} \qquad E(67) = \frac{\partial\ddot{x}}{\partial\psi_Y} \qquad E(68) = \frac{\partial\ddot{x}}{\partial M}$$

$$E(69) = \frac{\partial\ddot{x}}{\partial\epsilon_{\dot{M}}} \qquad E(70) = \frac{\partial\ddot{x}}{\partial\epsilon_I}$$

$$E(71) = \frac{\partial\ddot{y}}{\partial x} \qquad E(72) = \frac{\partial\ddot{y}}{\partial y} \qquad E(73) = \frac{\partial\ddot{y}}{\partial z}$$

$$E(74) = \frac{\partial\ddot{y}}{\partial x_{M/E}} \qquad E(75) = \frac{\partial\ddot{y}}{\partial y_{M/E}} \qquad E(76) = \frac{\partial\ddot{y}}{\partial z_{M/E}}$$

$E(77) = \dfrac{\partial \ddot{y}}{\partial \psi_P}$ $E(78) = \dfrac{\partial \ddot{y}}{\partial \psi_Y}$ $E(79) = \dfrac{\partial \ddot{y}}{\partial M}$

$E(80) = \dfrac{\partial \ddot{y}}{\partial \epsilon_{\dot{M}}}$ $E(81) = \dfrac{\partial \ddot{y}}{\partial \epsilon_I}$

$E(82) = \dfrac{\partial \ddot{z}}{\partial x}$ $E(83) = \dfrac{\partial \ddot{z}}{\partial y}$ $E(84) = \dfrac{\partial \ddot{z}}{\partial z}$

$E(85) = \dfrac{\partial \ddot{z}}{\partial x_{M/E}}$ $E(86) = \dfrac{\partial \ddot{z}}{\partial y_{M/E}}$ $E(87) = \dfrac{\partial \ddot{z}}{\partial z_{M/E}}$

$E(88) = \dfrac{\partial \ddot{z}}{\partial \psi_P}$ $E(89) = \dfrac{\partial \ddot{z}}{\partial \psi_Y}$ $E(90) = \dfrac{\partial \ddot{z}}{\partial M}$

$E(91) = \dfrac{\partial \ddot{z}}{\partial \epsilon_{\dot{M}}}$ $E(92) = \dfrac{\partial \ddot{z}}{\partial \epsilon_I}$

$E(93) = \ddot{x}$ $E(94) = \ddot{y}$ $E(95) = \ddot{z}$

$E(96) = \dddot{x}$ $E(97) = \dddot{y}$ $E(98) = \dddot{z}$

$E(99) = N$

$$E(100) = 3 \frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} \frac{x - x_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$$

$$E(101) = 3 \frac{\mu_M}{|\underline{R}_{V/E}-\underline{R}_{M/E}|^3} \frac{y - y_{M/E}}{|\underline{R}_{V/E}-\underline{R}_{M/E}|}$$

$$E(102) = 3 \frac{\mu_E}{|\underline{R}_{V/E}|^3} \frac{x}{|\underline{R}_{V/E}|} \qquad E(103) = 3 \frac{\mu_E}{|\underline{R}_{V/E}|^3} \frac{y}{|\underline{R}_{V/E}|}$$

$$E(104) = 3 \frac{\mu_M}{|\underline{R}_{M/E}|^3} \frac{x_{M/E}}{|\underline{R}_{M/E}|} \qquad E(105) = 3 \frac{\mu_M}{|\underline{R}_{M/E}|^3} \frac{y_{M/E}}{|\underline{R}_{M/E}|}$$

The detailed programing instructions are shown here. Note that X(I), P(I), and E(I) should be in common.

```
      E(20)  =  DSIN(X(7))
      E(21)  =  DSIN(X(8))
      E(22)  =  DCØS(X(7))
      E(23)  =  DCØS(X(8))
      E(24)  =  E(21)*E(20)
      E(25)  =  E(21)*E(22)
      E(26)  =  E(23)*E(20)
      E(27)  =  E(23)*E(22)
      IF |E(31)| > P(17) GØ TØ a
      N  =  1
      GØ TØ b
a     N = 1. DO + (DABS(E(31)))/P(17)
      E(99)  =  N
      E(31)  =  E(31)/E(99)
b     DØ n    L = 1, N
      E(1)  =  E(1) + E(31)
      DØ c   I = 1, 3
      E(I+13)  =  E(I+1) - E(I+7)
c     E(I+16)  =  P(I+69) + X(I+9)
      DØ d   I = 1, 13, 6
      E(I+27)  =  E(I+1)**2 + E(I+2)**2 + E(I+3)**2
```

```
d     E(I+28)  =  DSQRT(E(I+27))
      E(32)  =  E(31)**2/2.DO
      E(33)  =  E(32)*E(31)/3.DO
      E(30)  =  (P(66)/E(28))/E(29)
      E(36)  =  (P(67)/E(34))/E(35)
      E(42)  =  (P(67)/E(40))/E(41)
      E(37)  =  P(69)/X(9)
      E(38)  =  E(37)*(P(73) + X(13))
      E(39)  =  E(37)*E(19)
      DØ e   J = 1, 3
      E(J+45)  =  P(J+73)*E(26) + P(J+76)*E(27) + P(J+79)*E(21)
      DØ e   I = 1, 13, 6
      K  =  I+J
e     E(K+41)  =  E(K)/E(I+28)
      E(52)  =  (P(68)/E(34))/E(35)
      E(53)  =  E(42) + E(30)
      E(54)  =  -E(42) + E(36)
      E(58)  =  E(38)*E(19)
      E(59)  =  E(58)/X(9)
      DØ f   I = 1, 2
      E(I+99)  =  3. DO*E(42)*E(I+54)
      E(I+101) =  3. DO*E(30)*E(I+42)
```

```
f     E(I+103) =  3. DO*E(36)*E(I+48)
      DØ g   I = 1, 3
      E(I+59)  =  E(100)*E(I+54) + E(102)*E(I+42)
g     E(I+62)  =  -E(100)*E(I+54) -E(104)*E(I+48)
      DØ h   I = 1, 2
      E(I+71)  =  E(101)*E(I+55) + E(103)*E(I+43)
h     E(I+74)  =  -E(101)*E(I+55) -E(105)*E(I+49)
      E(71)  =  E(61)
      E(74)  =  E(64)
      DØ i   I = 1, 2
      J  = 57 + 3*I
      E(J)  =  E(J) - E(I+52)
      E(J+12)  =  E(J+12) - E(I+52)
      E(J+22)  =  E(J+2)
      E(J+23)  =  E(J+13)
i     E(J+24)  =  -E(J) - E(J+12)
      DØ j   I = 1, 3
      J  =  55 + 11*I
      E(J)  =  E(58)*(-P(I+73)*E(27) + P(I+76)*E(26))
      E(J+1)  =  E(58)*(P(I+73)*E(24) + P(I+76)*E(25) -P(I+79)*E(23))
      E(J+2)  =  E(I+45)*E(59)
      E(J+3)  =  -E(I+45)*E(38)
```

```
j     E(J+4)  =  -E(I+45)*E(39)
      DØ k   I = 1, 3
      J  =  49 + 11*I
      E(I+92)  =  -E(42)*E(I+13) -E(30)*E(I+1) -E(36)*E(I+7) -E(I+45)*E(58)
      E(I+95)  =  E(J)*E(5) + E(J+1)*E(6) + E(J+2)*E(7) + E(J+3)*E(11)
               + E(J+4)*E(12) + E(J+5)*E(13) + E(J+6)*E(17)
               + E(J+7)*E(18) + E(J+8)*E(19)
k     E(I+1)  =  E(I+1) + E(I+4)*E(31) + E(I+92)*E(32) + E(I+95)*E(33)
      DØ m   I = 1, 3
      E(I+10)  =  E(I+10) -E(52)*E(I+7)*E(31)
      E(I+7)  =  E(I+7)*(1.DO + E(52)*E(32)) + E(I+10)*E(31)
m     E(I+4)  =  E(I+4) + E(I+92)*E(31) + E(I+95)*E(32)
n     CØNTINUE
```

14.6 Programing Instructions for Blocks 1 and 2

Blocks 1 and 2 are the first two blocks in the overall block diagram shown in figure 5. The programing instructions for these two blocks are shown here. Based on results from future simulation studies, the equations in block 1 may have to be slightly modified. The logic for block 1 is included in flow chart 1 of the appendix.

comment BEGIN BLOCK 1

Zero F(I), X(I), YY(I), JX(I, J), P(I), Y(I)

Input the nonzero P(I) input constants. (If all P(I), including zero values, are read in, then omit zeroing P(I) above.)

Set descent (F(32) = 0), ascent (F(32) = 1) flag.

Read in the RNP matrix as shown below:

$$\begin{bmatrix} P(1) & P(2) & P(3) \\ P(4) & P(5) & P(6) \\ P(7) & P(8) & P(9) \end{bmatrix} = RNP$$

Obtain time of powered flight initiation from Mission Plan Table (MPT) and store in P(120)

comment INITIALIZE FOR DESCENT OR ASCENT .

IF $(F(32) \neq 0)$ GO TO yy

$P(15) = P(120) - P(124)$

IF$((T_{RL} - P(20))$. LT. $P(15))$ GO TO xx

IF $((T_{RL} - P(20))$. LT. $P(120))$ GO TO ww

$P(15) = P(120)$

$P(18) = (T_{RL} - P(20))_R$ [a]

STORE MPT LM MCI VECTOR IN P(I), I = 60, 65

GO TO zz

ww $P(15) = T_{RL} - P(20)$

xx $P(18) = (P(15) + P(119))_R$ [a]

---

[a]This step, though easily done, is very critical. If not done properly, the filter may get out of "sync" with the observation times and fail to take in any measurements. Note, also, that it is assumed that measurements are available exactly on the second mark. For example, 5-measurements-per-second data cannot be processed with time tags of 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, ... seconds.

CALL AEG. INTEGRATE MPT LM MCI VECTOR TO P(15).
STORE IN P(I), I = 60, 65.

GO TO zz

yy P(15) = P(120) - P(125)

COMPUTE MCT STATE VECTOR OF LAUNCH SITE. (See flow chart 1 in appendix.)

CALL ELVCNV. CONVERT MCT VECTOR TO MCI AND STORE MCI VECTOR IN P(I), I = 60, 65.

OBTAIN CSM MCI STATE VECTOR AT P(15) FROM VEHICLE EPHEMERIS. STORE IN P(I), I = 74, 79.

zz CALL ELVCNV. CONVERT LM MCI VECTOR TO ECI. STORE ECI VECTOR IN X(I), I = 1, 6.

IF F(32) ≠ 0 GØ TØ b

DØ a I = 1, 3

E(I) = P(I+59)

E(I+3) = -P(I+62)

JX(I, I) = P(102)

a JX(I+3, I+3) = P(103)

JX(7, 7) = P(104)

JX(8, 8) = P(104)

JX(9, 9) = P(105)

X(9) = P(117)

GØ TØ d

```
b         DØ c   I = 1, 3
          E(I)  =  P(I+73)
          E(I+3)  =  P(I+76)
          JX(I, I)  =  P(106)
c         JX(I+3, I+3)  =  P(107)
          JX(7, 7)  =  P(108)
          JX(8, 8)  =  P(108)
          JX(9, 9)  =  P(109)
          JX(12, 12)  =  P(123)
          P(90)  =  P(123)
          P(72)  =  P(121)
          F(34)  =  1
          X(7)  =  1.5625 DO
          X(9)  =  P(118)
d         JX(10, 10)  =  P(88)
          JX(11, 11)  =  P(89)
          JX(13, 13)  =  P(91)
comment   CALCULATE C MATRIX
          E(7)  =  DSQRT(P(60)**2 + P(61)**2 + P(62)**2)
          DØ e   I = 1, 3
e         P(I + 73)  =  P(I+59)/E(7)
          E(8)  =  E(2)*E(6) -E(3)*E(5)
          E(9)  =  E(3)*E(4) -E(1)*E(6)
```

```
        E(10)  =  E(1)*E(5) -E(2)*E(4)
        P(77)  =  E(9)*P(76) -E(10)*P(75)
        P(78)  =  E(10)*P(74) -E(8)*P(76)
        P(79)  =  E(8)*P(75) -E(9)*P(74)
        E(11)  =  DSQRT(P(77)**2 + P(78)**2 + P(79)**2)
        DØ f   I = 77, 79
f       P(I)  =  P(I)/E(11)
        P(80)  =  P(75)*P(79) -P(76)*P(78)
        P(81)  =  P(76)*P(77) -P(74)*P(79)
        P(82)  =  P(74)*P(78) -P(75)*P(77)
        DØ g   I = 1, 6
g       P(I+59)  =  X(I) -P(I+59)
comment BEGIN BLOCK 2
        CALL TRANST
        P(16) = E(86)
```

14.7 Programing Instructions for Blocks 3, 4, 5, and 6

These blocks constitute the dynamics portion of the program. Based on results from future simulation studies, the equations in block 3 may have to be slightly modified. The logic for a restart procedure and the logic for block 3 are included in flow charts 2 and 3 respectively of the appendix.

Just before the instruction "IF P(18) + P(20) > current time WAIT" is the ideal time to interrupt this program to perform other shared operations with the computer. This instruction forces the program to be synchronized with real time. Because the program cycle will execute "faster" than real time, there always will be a pause at this instruction to allow real time to catch up.

```
a         CØNTINUE
          IF (P(18) + P(20)) > T_RL (current time in hours) WAIT
comment   BEGIN BLOCK 3
          IF (F(35) ≠ 0) GØ TØ aaa
          IF F(34) ≠ 0  GØ TØ ab
          IF P(15) < P(120)  GØ TØ ag
comment   INITIALIZE THRUST ANGLE FØR DESCENT
          DØ aa  I = 1, 3
aa        E(I)  =  P(I+62) - X(I+3)
aaa       F(35) = 0
          E(4)  =  P(74)*E(1) + P(75)*E(2) + P(76)*E(3)
          E(5)  =  P(77)*E(1) + P(78)*E(2) + P(79)*E(3)
          E(6)  =  P(80)*E(1) + P(81)*E(2) + P(82)*E(3)
          E(7)  =  DSQRT(E(4)**2 + E(5)**2)
          X(7)  =  DATAN2(E(4), E(5))
          X(8)  =  DATAN2(E(6), E(7))
          JX(12, 12)  =  P(123)
          P(90)  =  P(123)
          F(34)  =  1
comment   SET DATA DRØPØUT FLAG
ab        F(31)  =  1
          IF (F(9) + F(10) + F(11) + F(12)  =  0)  SET F(31)  =  0
comment   SET MASS FLØW RATE VARIANCE
          IF (F(31) - F(33))  ac, af, ad
```

```
ac        P(90)  =  P(123)
          GØ TØ ae
ad        JX(12, 12)  =  P(122)
          P(90)  =  P(122)
ae        F(33)  =  F(31)
comment   SET NØMINAL VALUE OF MDØT TØ BEST ESTIMATE ØF MDØT
af        P(72)  =  P(72) + X(12)
          X(12)  =  0.  DO
ag        CØNTINUE
comment   BEGIN BLOCK 4
          E(31)  =  P(16)
          E(1)  =  P(15)
          DØ b   I = 1, 6
          E(I+1)  =  X(I)
b         E(I+7)  =  P(I+59)
          CALL TRAJ
          P(15)  =  E(1)
          DØ c   I = 1, 6
          X(I)  =  E(I+1)
c         P(I+59)  =  E(I+7)
comment   SET ΔT*
          IF F(34)  =  0  GØ TØ d
```

```
        M(4, 1)  =  P(16)
        GØ TØ e
d       M(4, 1)  =  0. DO
e       DØ f   I = 7, 9
f       X(I)  =  X(I) + E(I+10)*M(4, 1)
        DØ g   I = 10, 13
        X(I)  =  X(I)*P(I)
g       X(I+4)  =  X(I+4)*P(14)
comment BEGIN BLOCK 5
        DØ h   J = 1, 3
        K  =  48 + 11*J
        DØ h   I = 1, 3
        K  =  K+1
        M(J, I)  =  E(K)*P(16)
        M(J, I+6)  =  E(K+6)*P(16)
h       M(J, I+11)  =  E(K+9)*P(16)
comment BEGIN BLOCK 6
        DØ k   I = 1, 13
        DØ i   J = 1, 3
        A(I, J)  =  JX(I, J) + JX(I, J+3)*P(16)
i       A(I, J+3)  =  JX(I, 1)*M(J, 1) + JX(I, 2)*M(J, 2)
                  + JX(I, 3)*M(J,3) + JX(I, 7)*M(J, 7)
                  + JX(I, 8)*M(J,8) + JX(I, 9)*M(J, 9)
                  + JX(I, 12)*M(J, 12) + JX(I, 13)*M(J, 13)
                  + JX(I, J+3)
```

```
        DØ j   J = 7, 9
j       A(I, J)  =  JX(I, J) + JX(I, J+3)*M(4,1)
        DØ k   J = 10, 13
k       A(I, J)  =  JX(I, J)*P(J)
        DØ n   I = 1, 3
        DØ m   J = I, 13
m       JX(I, J)  =  A(I, J) + A(I+3, J)*P(16)
        L  =  I+ 3
        DØ n   J = L, 13
n       JX(L, J)  =  M(I, 1)*A(1, J) + M(I, 2)*A(2, J) + M(I,3)*A(3, J)
                     + M(I, 7)*A(7, J) + M(I, 8)*A(8, J) + M(I, 9)*A(9, J)
                     + M(I, 12)*A(12, J) + M(I, 13)*A(13, J) + A(L, J)
        DØ o   I = 7, 9
        DØ o   J = I, 13
o       JX(I, J)  =  A(I, J) + A(I+3, J)*M(4, 1)
        DØ p   I = 10, 13
        DØ p   J = I, 13
p       JX(I, J)  =  P(I)*A(I, J)
        DØ q   I = 2, 13
        L  =  I-1
        DØ q   J = 1, L
q       JX(I, J)  =  JX(J, I)
        DØ r   J = 1, 13
        DØ r   I = 1, 4
        A(I, J)  =  P(14)*JX(I+13, J)
```

```
r        A(I+4, J)  =  JX(I+17, J)
         DØ u   I = 1, 8
         DØ s   J = 1, 3
         JX(I+13, J)  =  A(I, J) + A(I, J+3)*P(16)
s        JX(I+13, J+3)  =  A(I, 1)*M(J, 1) + A(I, 2)*M(J, 2)
                  + A(I, 3)*M(J, 3) + A(I, 7)*M(J, 7)
                  + A(I, 8)*M(J, 8) + A(I, 9)*M(J, 9)
                  + A(I, 12)*M(J, 12) + A(I, 13)*M(J, 13)
                  + A(I, J+3)
         DØ t   J = 7, 9
t        JX(I+13, J)  =  A(I, J) + A(I, J+3)*M(4, 1)
         DØ u   J = 10, 13
u        JX(I+13, J)  =  A(I, J)*P(J)
         E(1)  =  P(14)**2
         DØ v   I = 14, 17
         DØ v   J = I, 17
         JX(I, J)  =  E(1)*JX(I, J)
v        JX(J, I)  =  JX(I, J)
         DØ w   I = 18, 21
         DØ w   J = 14, 17
         JX(I, J)  =  P(14)*JX(I, J)
w        JX(J, I)  =  JX(I, J)
comment  ADD R
         DØ x   I = 4, 6
```

```
x        JX(I, I)  =  JX(I, I) + P(92)
         DØ y   I = 10, 13
         JX(I, I)  =  JX(I, I) + P(I+73)*P(I+78)
y        JX(I+4, I+4)  =  JX(I+4, I+4) + P(87)*P(I+38)
```

14.8 Programing Instructions for Blocks 7 through 14

The check on |F(1)| + |F(2)| + |F(3)| + |F(4)|, shown functionally between blocks 9 and 10 in the overall block diagram, is made by equivalent checks at the beginning of block 10.

```
comment  BEGIN BLOCK 7

         M = 0

         DØ g   L = 1, 4

         IF there is a good[a] cycle count for the Lth station in the
         interval P(18) -1.D-5 to P(18) + 1.D-5 hours GØ TØ a

         P(L+109) = 0.D0

         M = M+1

         F(L+4)  =  0

         F(L)  =  0

         GØ TØ f
```

[a]The word good means that the tracking station has attached a good label to the data.

a Store count in YY(L)

Set F(L+12) = 0 for 3-way Doppler.[b]

Set F(L+12) = 1 for 2-way Doppler.[b]

Load station ID number into F(L+16).[b]

Load b_L, a priori "rate bias" in cycles/hour, into P(L+51).

```
    E(94)  =  YY(L) -P(L+55)
    IF (E(94) ≤ P(114)) GØ TØ b
    IF (E(94) ≥ P(115)) GØ TØ b
    IF (DABS(E(94) -P(L+109))) ≥ P(116)  GØ TØ b
    F(L+4)  =  1
    GØ TØ c
b   F(L+4)  =  0
c   P(L+109)  =  E(94)
    IF (F(L+4) -F(L+8)) > 0  GØ TØ d
    F(L)  =  0
    GØ TØ e
d   F(L)  =  1
e   IF F(L+16)  =  F(L+21)  GØ TØ f
```

[b]The information concerning the flag, F(L + 12), and the station ID number, F(L + 16), is contained in the tracking station input word.

Load geodetic longitude of station F(L+16) into P(L+29).[a]

Load $R_E \cos \varphi' + H \cos \varphi$ for station F(L+16) into P(L+34).[a]

Load $R_E \sin \varphi' + H \sin \varphi$ for station F(L+16) into P(L+39).[a]

```
            F(L)     =  -1
f           P(L+55)  =  YY(L)
g           F(L+8)   =  F(L+4)
            IF M = 4 GØ TØ i
```

Load transmitter ID into F(21).

```
            IF F(21)  =  F(26)  GØ TØ i
```

Load geodetic longitude of station F(21) into P(34).*

Load $R_E \cos \varphi' + H \cos \varphi$ for station F(21) into P(39).*

Load $R_E \sin \varphi' + H \sin \varphi$ for station F(21) into P(44).*

```
            DØ h   I = 1, 4
h           F(I)     =  -1
i           DØ j   I = 1, 5
j           F(I+21)  =  F(I+16)
comment     BEGIN BLOCK 8
            CALL TRANST
            P(16)  =  E(86) + P(19)
```

---

[a]These quantities may be loaded directly from the station characteristics table.

```
comment   BEGIN BLOCK 9
          DØ o   L = 1, 4
          IF F(L+4)  =  0   GØ TØ m
          K  =  L+33
          M(L, L+13)  =  P(18) -P(L+96)
          M(L, L+17)  =  -1.DO
          DØ k   J = 1, 3
          K  =  K+4
          M(L, J)  =  P(93)*(E(K)/E(L+49) + E(K+32)/E(L+81))
k         M(L, J+3)  =  E(L+85)*M(L, J)
          GØ TØ o
m         DØ n   I = 1, 6
n         M(L, I)  =  0. DO
          M(L, L+13)  =  0. DO
          M(L, L+17)  =  0. DO
o         CONTINUE
comment   BEGIN BLOCK 10
          DØ v   L = 1, 4
          IF (F(L))  =  0   GØ TØ v
          Y(L+17)  =  P(93)*(E(L+49) + E(L+81)) -YY(L)
          P(L+96)  =  P(18)
          DØ p   J = 1, 21
p         JX(L+17, J)  =  0. DO
          DØ q   J = 14, 21
```

```
q       JX(J, L+17)  =  0. DO
        IF (F(L)) > 0   GØ TØ v
        IF (F(L+12))  =  0  GØ TØ r
        P(L+47)  =  P(47)
        GØ TØ s
r       P(L+47)  =  P(46)
s       X(L+13)  =  0. DO
        DØ t   J = 1, 21
t       JX(L+13, J)  =  0. DO
        DØ u   J = 14, 21
u       JX(J, L+13)  =  0. DO
        JX(L+13, L+13)  =  P(L+47)
v       CØNTINUE
        DØ z   L = 1, 4
        IF (F(L))  =  0  GØ TØ z
        DØ w   J = 1, 21
w       JX(L+17, J)  =  M(L, 1)*JX(J, 1) + M(L, 2)*JX(J, 2)
                      + M(L, 3)*JX(J, 3) + M(L, 4)*JX(J, 4)
                      + M(L, 5)*JX(J, 5) + M(L, 6)*JX(J, 6)
        DØ x   J = 14, 21
x       JX(J, L+17)  =  JX(L+17, J)
```

```
        JX(L+17, L+17)  =  JX(L+17, L+17) + P(101)
        M(L, L+13)  =  0. D0
        M(L, L+17)  =  0. D0
        DØ y   J = 1, 6
y       M(L, J)  =  0. D0
z       CØNTINUE
comment BEGIN BLOCK 11
        DØ aa   L = 1, 4
        Y(L)  = YY(L) -P(93)*(E(L+49) + E(L+81)) -(P(94) + P(L+51)
              + X(L+13))*(P(18) -P(L+96)) + X(L+17)
        IF (F(L+4)) = 0  SET Y(L)  = 0. D0
aa      CØNTINUE
        P(18)  =  P(18) + P(19)
        M  =  0
comment BEGIN BLOCK 12
comment FORM D = J(M TRANSPOSE)
ab      DØ ac   I = 1, 21
        DØ ac   J = 1, 4
ac      D(I, J)  =  JX(I, 1)*M(J, 1) + JX(I, 2)*M(J, 2) + JX(I, 3)*M(J,3)
                 + JX(I, 4)*M(J, 4) + JX(I, 5)*M(J, 5) + JX(I, 6)*M(J,6)
                 + JX(J+13, I)*M(J,J+13) + JX(J+17,I)*M(J,J+17)
comment FORM H = MD + W
        DØ ae   I = 1, 4
        DØ ad   J = 1, I
```

```
ad        H(I, J)  =  M(I, 1)*D(1, J) + M(I, 2)*D(2, J) + M(I, 3)*D(3, J)
                    + M(I, 4)*D(4, J) + M(I, 5)*D(5, J) + M(I, 6)*D(6, J)
                    + M(I, I+13)*D(I+13, J) + M(I, I+17)*D(I+17, J)

ae        H(I, I)  =  H(I, I) + P(101)

comment   FORM H  =  H INVERSE

          H(1, 1)  =  1.DO/H(1, 1)

          DØ aj   J = 1, 3

          L  =  J+1

          DØ af   I = 1,J

          H(I, L)  =  0. DO

          DØ af   K = 1, J

af        H(I, L)  =  H(I, L) - H(L, K)*H(I, K)

          DØ ag   K = 1, J

ag        H(L, L)  =  H(L, L) + H(L, K)*H(K, L)

          H(L, L)  =  1.DO/H(L, L)

          DØ ah   I = 1, J

ah        H(L, I)  =  H(I, L)*H(L, L)

          DØ ai   I = 1, J

          DØ ai   K = I, J

          H(I, K)  =  H(I, K) + H(I, L)*H(L, K)

ai        H(K, I)  =  H(I, K)

          DØ aj   I = 1, J

aj        H(I, L)  =  H(L, I)
```

```
comment  FORM B = DH
         DØ ak   I = 1, 21
         DØ ak   J = 1, 4
ak       B(I, J)  =  D(I, 1)*H(1, J) + D(I, 2)*H(2, J) + D(I, 3)*H(3, J)
                     + D(I, 4)*H(4, J)
         IF M ≠ 0  GØ TØ ap

comment  BEGIN BLOCK 13
         DØ an   L = 1, 4
         IF (F(L+8))  =  0  GØ TØ an
         K  =  L+13
         H(L, 1)  =  DABS(P(L+51) + X(K) + B(K, 1)*Y(1) + B(K, 2)*Y(2)
                          + B(K, 3)*Y(3) + B(K, 4)*Y(4))
         IF (H(L, 1) < P(45))  GØ TØ an
         F(L+8)  =  0
         DØ am   J = 1, 6
am       M(L, J)  =  0. DO
         M(L, K)  =  0. DO
         M(L, L+17)  =  0. DO
an       CØNTINUE
         DØ ao   L = 1, 4
         IF (F(L+4)  =  0) GØ TØ ao
         IF (H(L, 1) < P(45))  GØ TØ ao
         Y(L)  =  0. DO
```

```
ao        CØNTINUE
          IF (F(5) + F(6) + F(7) + F(8) - F(9) -F(10) -F(11) -F(12)) = 0
              GØ TO ap
          M  =  1
          GØ TØ ab
comment   BEGIN BLOCK 14
ap        CØNTINUE
comment   CØRRECT STATE VECTØR
          DØ aq   I = 1, 21
aq        X(I)  =  X(I) + B(I, 1)*Y(1) + B(I, 2)*Y(2) + B(I, 3)*Y(3)
                 + B(I, 4)*Y(4)

comment   CØRRECT STATE ERRØR CØVARIANCE MATRIX
          DØ ar   I = 1, 13
          DØ ar   J = I, 13
          JX(I, J)  =  JX(I, J) -B(I, 1)*D(J, 1) -B(I, 2)*D(J, 2)
                       -B(I, 3)*D(J, 3) -B(I, 4)*D(J, 4)
ar        JX(J, I)  =  JX(I, J)
          DØ at   I = 14, 21
          DØ as   J = 1, 13
as        JX(I, J)  =  JX(I, J) -B(I, 1)*D(J, 1) -B(I, 2)*D(J, 2)
                       -B(I, 3)*D(J, 3) -B(I, 4)*D(J, 4)
          DØ at   J = I, 21
          JX(I, J)  =  JX(I, J) -B(I, 1)*D(J, 1) -B(I, 2)*D(J, 2)
                       -B(I,3)*D(J, 3) -B(I, 4)*D(J, 4)
at        JX(J, I)  =  JX(I, J)
```

14.9 Programing Instructions for Blocks 15 and 16

These two blocks provide the position and velocity of the LM in selenocentric, MNBY coordinates with units of earth radii and earth radii per hour. The time tag of these vectors is stored in P(23). The position and velocity vectors are stored in P(I + 23) I = 1, 6.

The desired time, T_D, of the state vector is loaded into P(21) anywhere in the program prior to this time. The program will generate a state vector for this particular time each time there is a change in the value of T_D. For example, the program may be interrupted after block 14 to load T_D into P(21). Then, after block 16, the state for this time may be read out.

```
          IF (P(21))  =  0.DO  GØ TØ c
          IF (P(21))  =  P(22)  GØ TØ c
comment   BEGIN BLOCK 15
          E(31)  =  P(21) - P(15)
          E(1)  =  P(15)
          DØ a   I = 1, 6
          E(I+1)  =  X(I)
a         E(I+7)  =  P(I+59)
          P(22)  =  P(21)
          CALL TRAJ
comment   BEGIN BLOCK 16
          P(23)  =  P(21)
          DØ b   I = 2, 7
b         P(I+22)  =  E(I) - E(I+6)
c         CØNTINUE
          GØ TØ "continue statement preceding block 3".
```

15.0 CONTROLS AND DISPLAYS

General descriptions are presented of the various controls and displays used in the program. More detailed descriptions can be found in reference 2.

15.1 Manual Entry Device (MED)

Ascent - descent sites[a] - The ascent-descent sites device specifies up to four sites to be processed in the high-speed mode. This MED also allows the controller to change any of the sites being processed and to reinitialize the data tables for all trackers if a change in the identity of the two-way site occurs.

15.2 Push Button Indicators

1. Stop PBI - The stop PBI allows the controller to terminate program processing.

2. Station edit[a] - The group of four Station-edit PBI's determines which sites are to be processed by the powered-flight data processor. The ability exists to edit sites in or out of the program processing.

3. All except PBI[a] - The All-except PBI informs the program to process the tracking data from all stations with the exception of the station whose edit PBI also was entered. If the all-except PBI is entered alone, the data of all sites is inserted for program processing.

4. PGNCS restart - The PGNCS-restart PBI enables the program processing to restart and specifies that the two most current PGNCS vectors be used in the restart procedure.

5. AGS restart - The AGS-restart PBI enables the program processing to restart and specifies that the two most current AGS vectors be used in the restart procedure.

15.3 Displays

1. LM descent/abort vector selection - The LM descent/abort vector selection display provides an analog trend curve of $\dot{W}$ versus $\dot{h}$ for PGNCS, AGS, and MSFN during the lunar descent phase or descent abort phase. The quantity $\dot{W}$ is the magnitude of the out-of-plane velocity vector component as computed by PGNCS, AGS, and MSFN.

---

[a]This control was previously designed for the RTCC (ref. 1) and is currently a physical part of the RTCC control system.

$$\dot{W}_i = \vec{V}_{LM/i} \cdot \left(\frac{\vec{R}_{LS} \times \vec{R}_{LM}}{|\vec{R}_{LS} \times \vec{R}_{LM}|} \right)_{t_{IGN}}$$

where $\vec{V}_{LM/i}$ = MCI velocity vector of LM at t_i

t_{IGN} = LM engine-on-time for powered lunar descent

$\vec{R}_{LS}$ = position vector of landing site at t_{IGN}

$\vec{R}_{LM}$ = position vector of LM at t_{IGN}

The value $\dot{h}$ is the altitude rate at which the LM approaches the lunar surface as computed by each of the three systems.

In addition to the three analog curves, digital quantities are also displayed which include values of pitch angle, yaw angle, LM mass, and LM mass flow rate as computed by the MSFN high-speed processor; altitude above the lunar surface as computed by MSFN, PGNCS, and AGS; tracker ID's for the sites being processed; and numerical values and indicators which describe the status of the tracking sites being processed. For more details, see reference 2.

2. The LM ascent vector selection - The LM ascent vector selection display provides an analog trend curve of $\dot{W}$ versus $|\vec{V}|$ for PGNCS, AGS, and MSFN during the lunar ascent phase. The quantity $|\vec{V}|$ is the magnitude of the LM velocity vector as computed by PGNCS, AGS, and MSFN. As for descent, $\dot{W}$ is the magnitude of the out-of-plane component of the velocity vector as determined by each of the three systems, but it is computed in a different method than is used for descent.

$$\dot{W}_i = \vec{V}_{LM/i} \cdot \left(\frac{\dot{\vec{R}}_{LM} \times \vec{R}_{LM}}{|\dot{\vec{R}}_{LM} \times \vec{R}_{LM}|} \right)_{t_{INS}}$$

where $\vec{V}_{LM/i}$ = MCI velocity vector of LM at t_i

t_{INS} = predicted time of LM insertion into lunar orbit

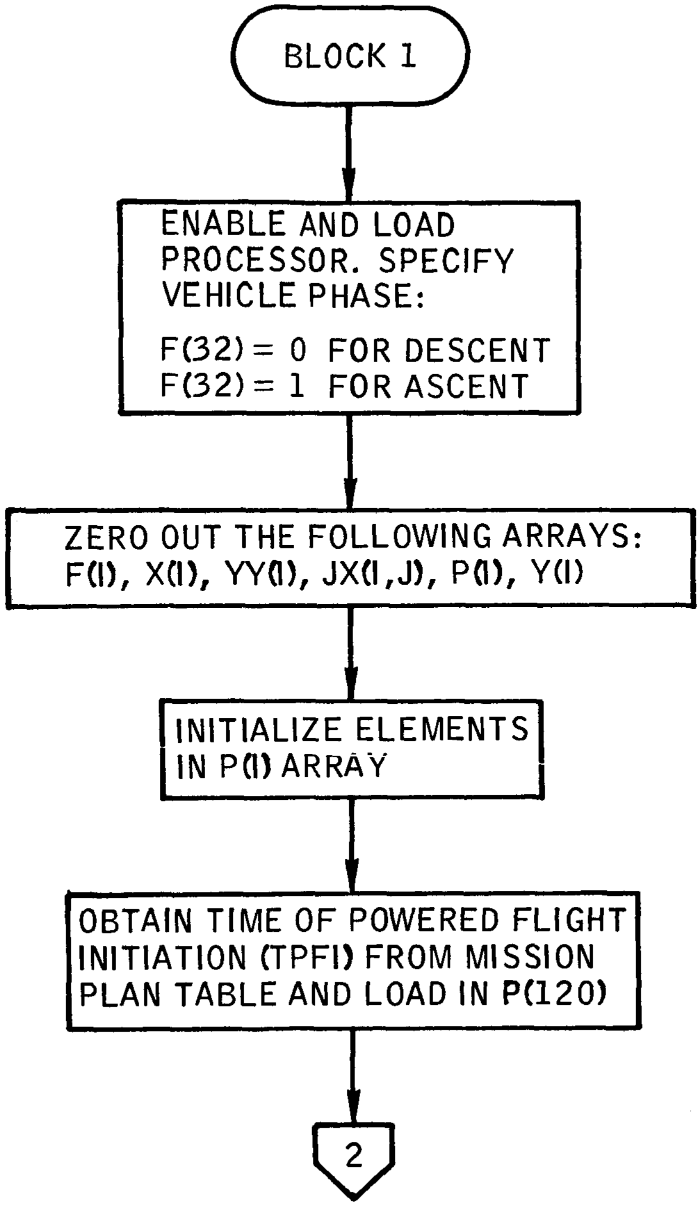
$\vec{R}_{LM}$ = predicted LM position vector at t_{INS}

$\dot{\vec{R}}_{LM}$ = predicted LM velocity vector at t_{INS}

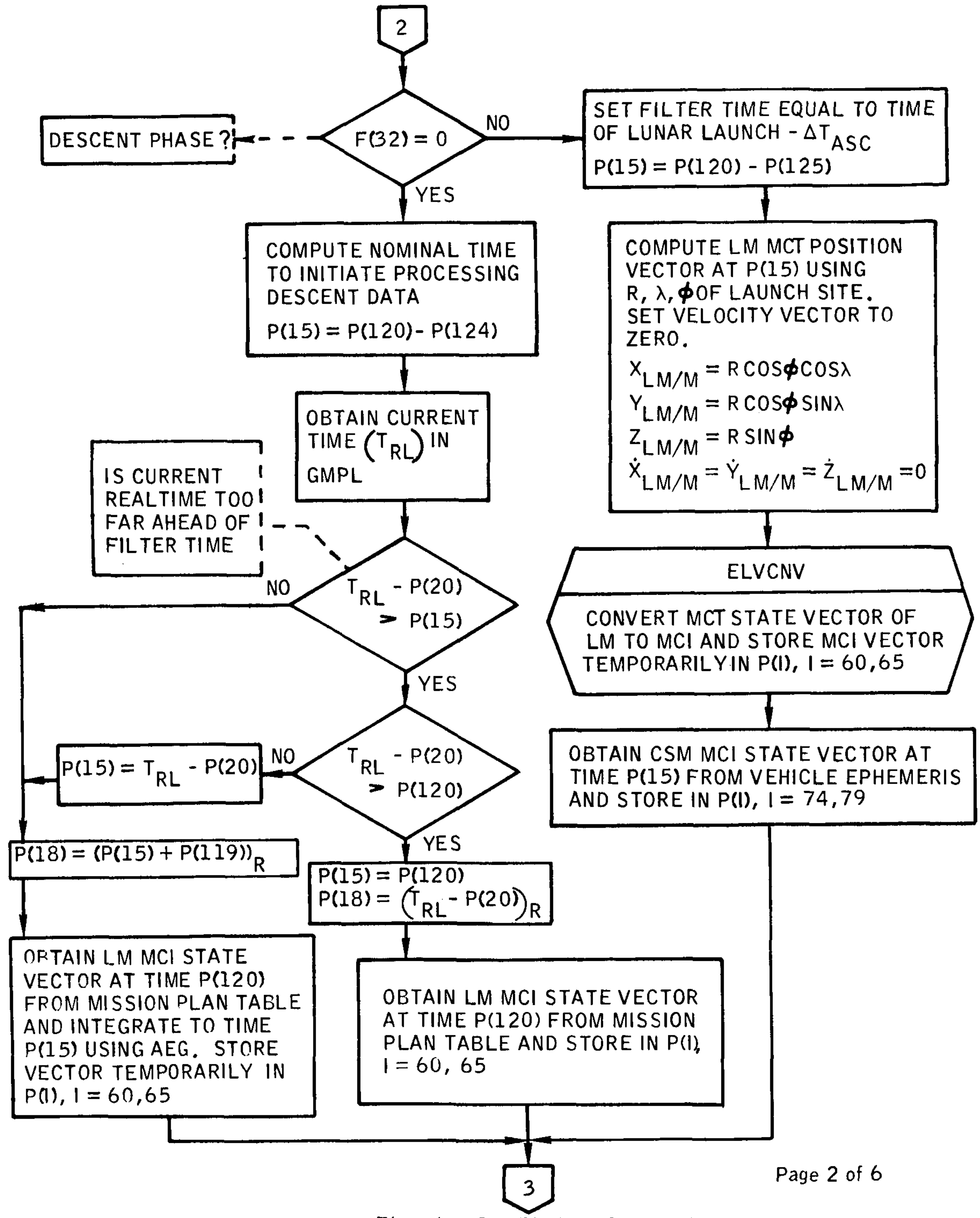
For more details, see reference 2. The same digital quantities are displayed for ascent as are for descent.

APPENDIX

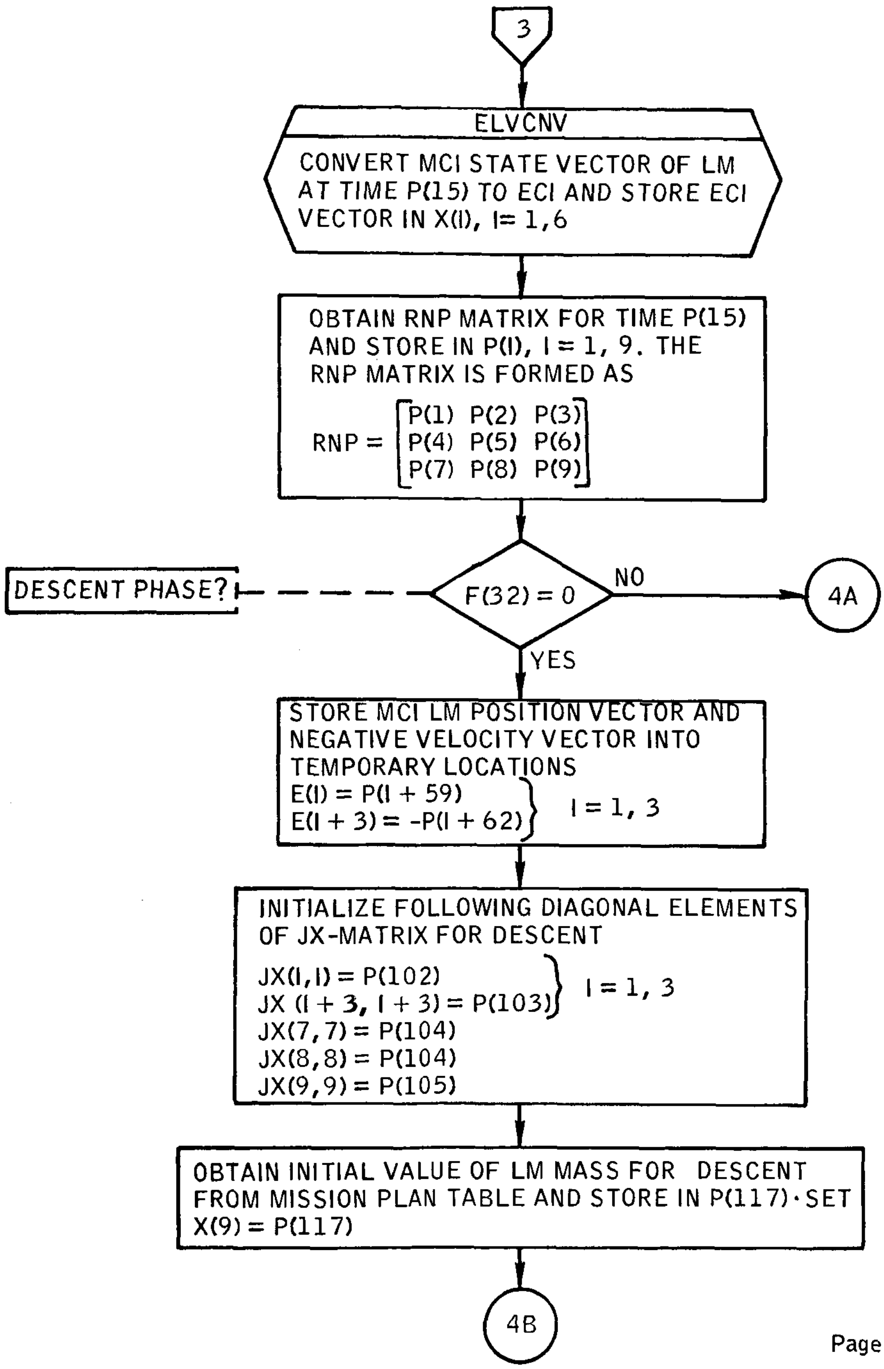
DETAILED FLOW CHARTS FOR BLOCK 1, RESTART, AND BLOCK 3



Flow chart 1.- Block I.



Flow chart 1.- Block I - Continued.



Page 3 of 6

Flow chart 1.- Block I - Continued.

4A

STORE CSM MCI STATE VECTOR IN TEMPORARY LOCATIONS
E(I) = P(I + 73)   I = 1, 6

INITIALIZE FOLLOWING DIAGONAL
ELEMENTS OF JX-MATRIX FOR ASCENT
$\left.\begin{array}{l} JX(I,I) = P(106) \\ JX(I+3,\ I+3) = P(107) \end{array}\right\}$ I = 1,3
JX(7,7) = P(108)
JX(8,8) = P(108)
JX(9,9) = P(109)

OBTAIN INITIAL VALUE OF MASS (M) FOR ASCENT FROM
MISSION PLAN TABLE AND STORE IN P(118)

INITIALIZE $\sigma^2_{\epsilon\dot{M}}$, $\dot{M}_N$, M FOR ASCENT
P(90) = P(123)
P(72) = P(121)
X(9) = P(118)

INITIALIZE VALUE OF PITCH ANGLE FOR ASCENT
X(7) = 1.5625 D0

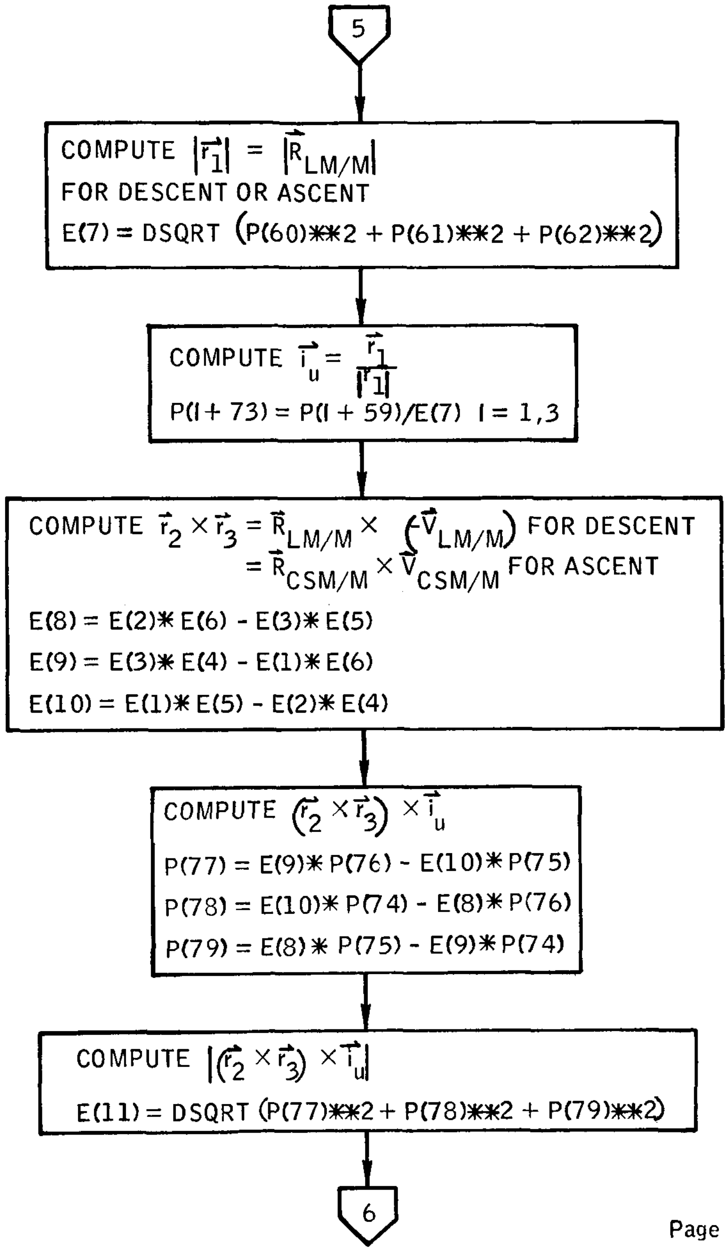
SET FLAG INDICATING VEHICLE NOT IN FREE FLIGHT
PHASE PRIOR TO POWER DESCENT: F(34) = 1

4B

INITIALIZE FOLLOWING ELEMENTS OF JX-MATRIX
JX(10,10) = P(88)
JX(11,11) = P(89)
JX(13,13) = P(91)

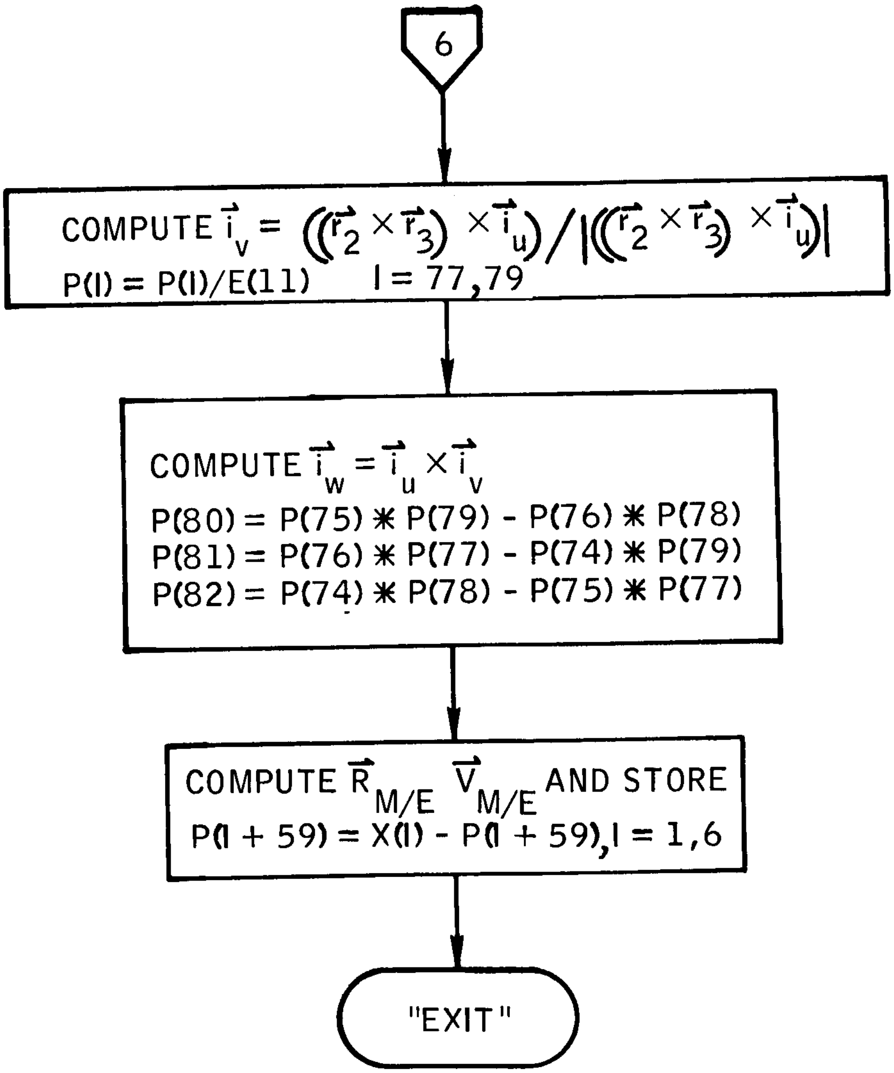
5

Flow chart 1.- Block I - Continued.

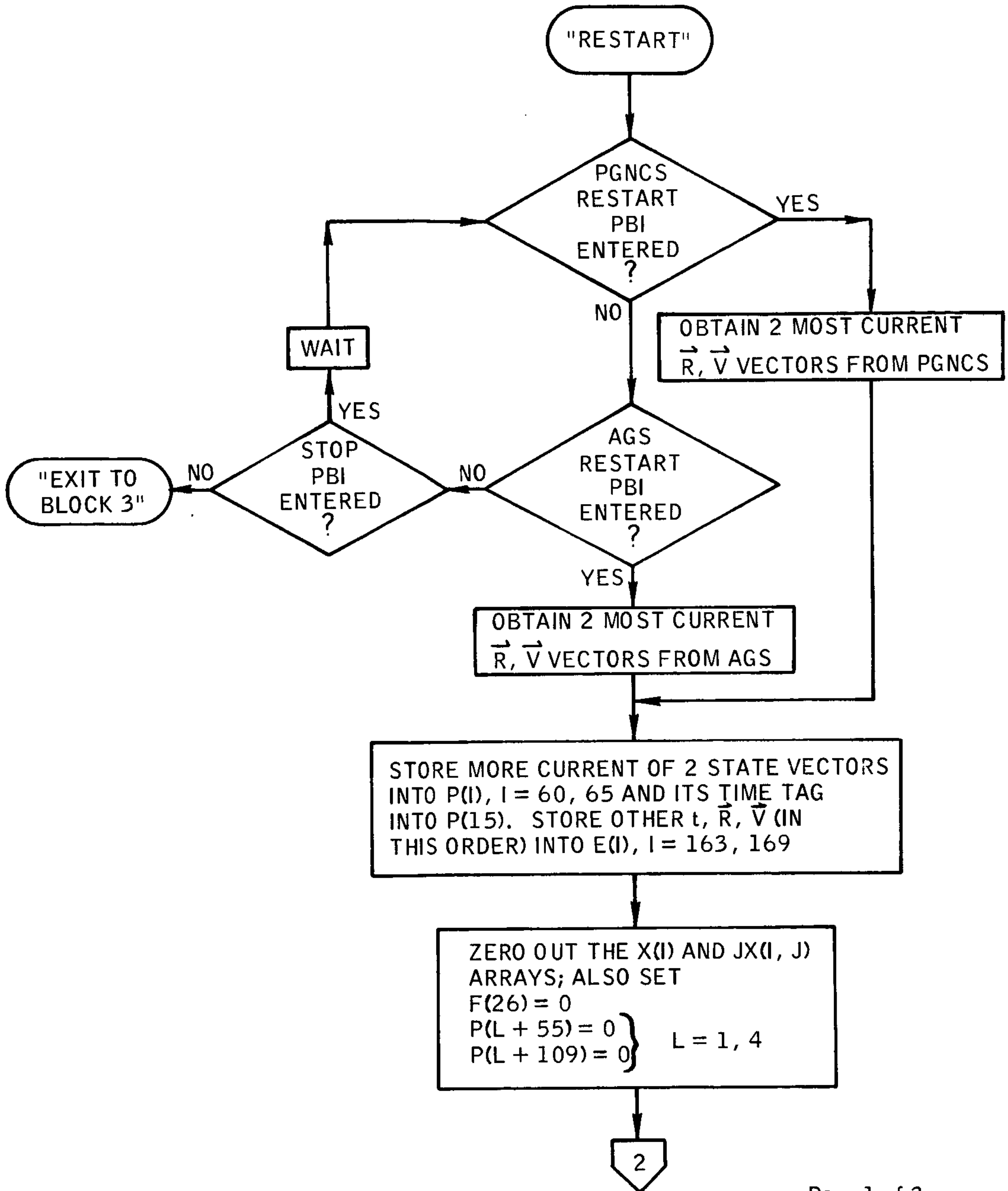


Page 5 of 6

Flow chart 1.- Block I - Continued.

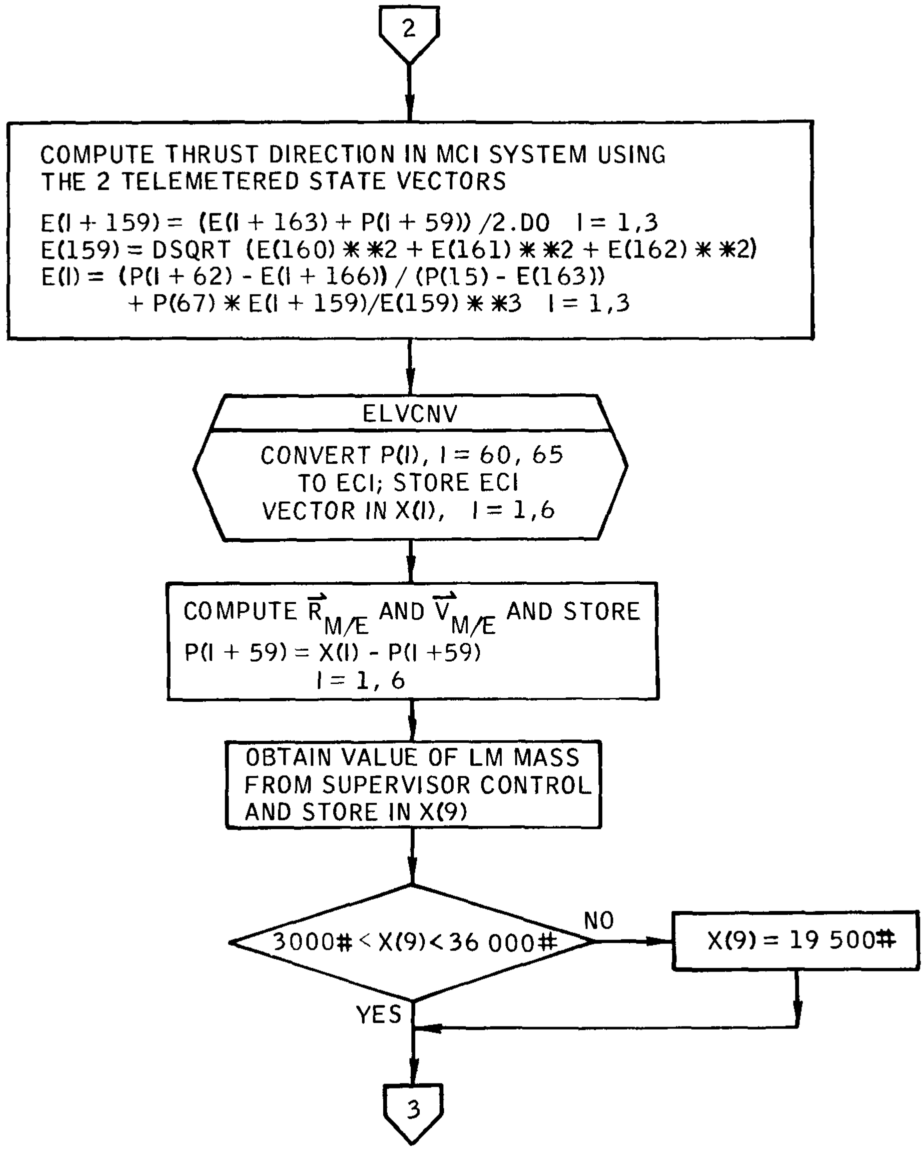


Flow chart 1.- Block I - Concluded.



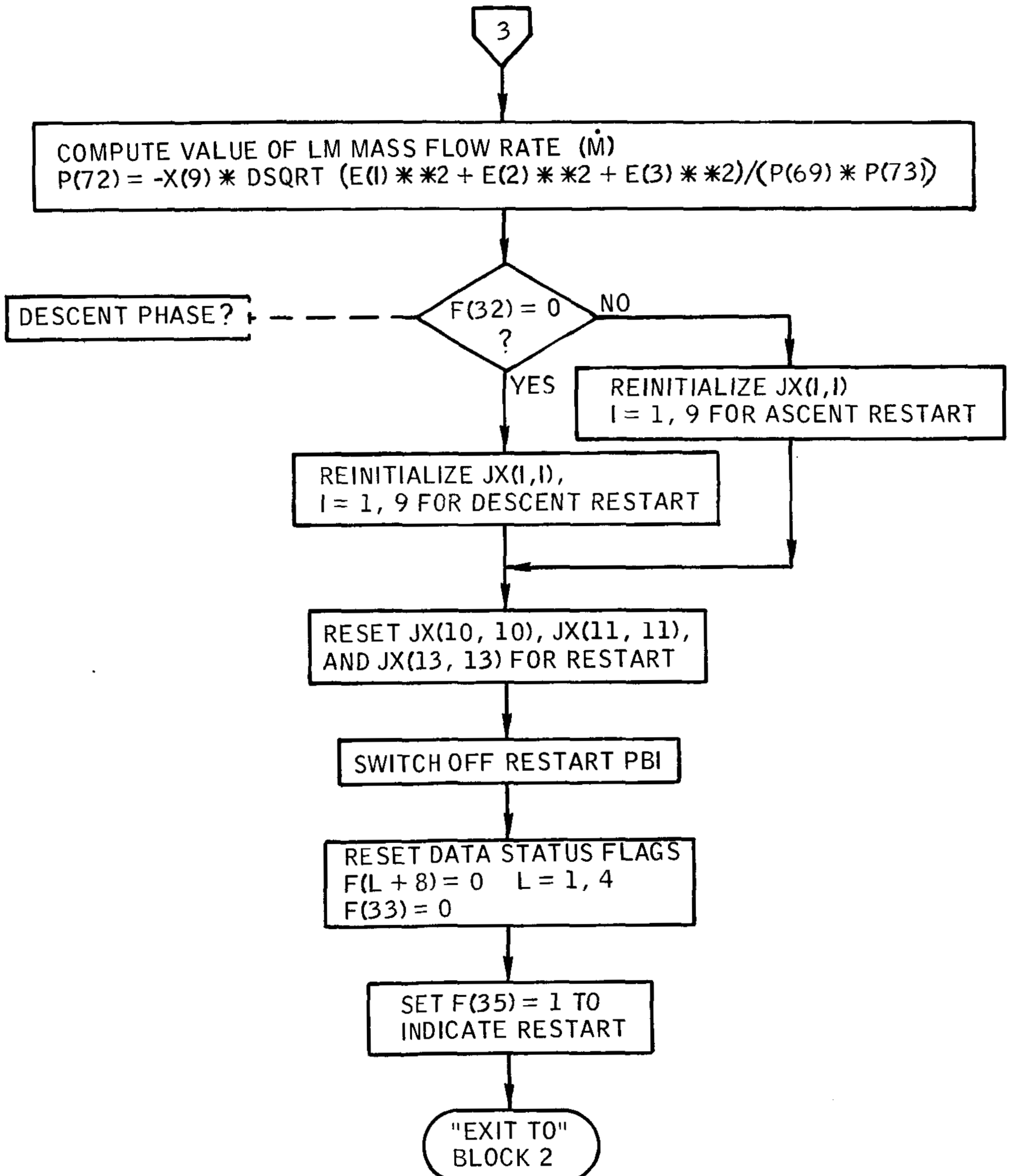
Page 1 of 3

Flow chart 2.- Restart.

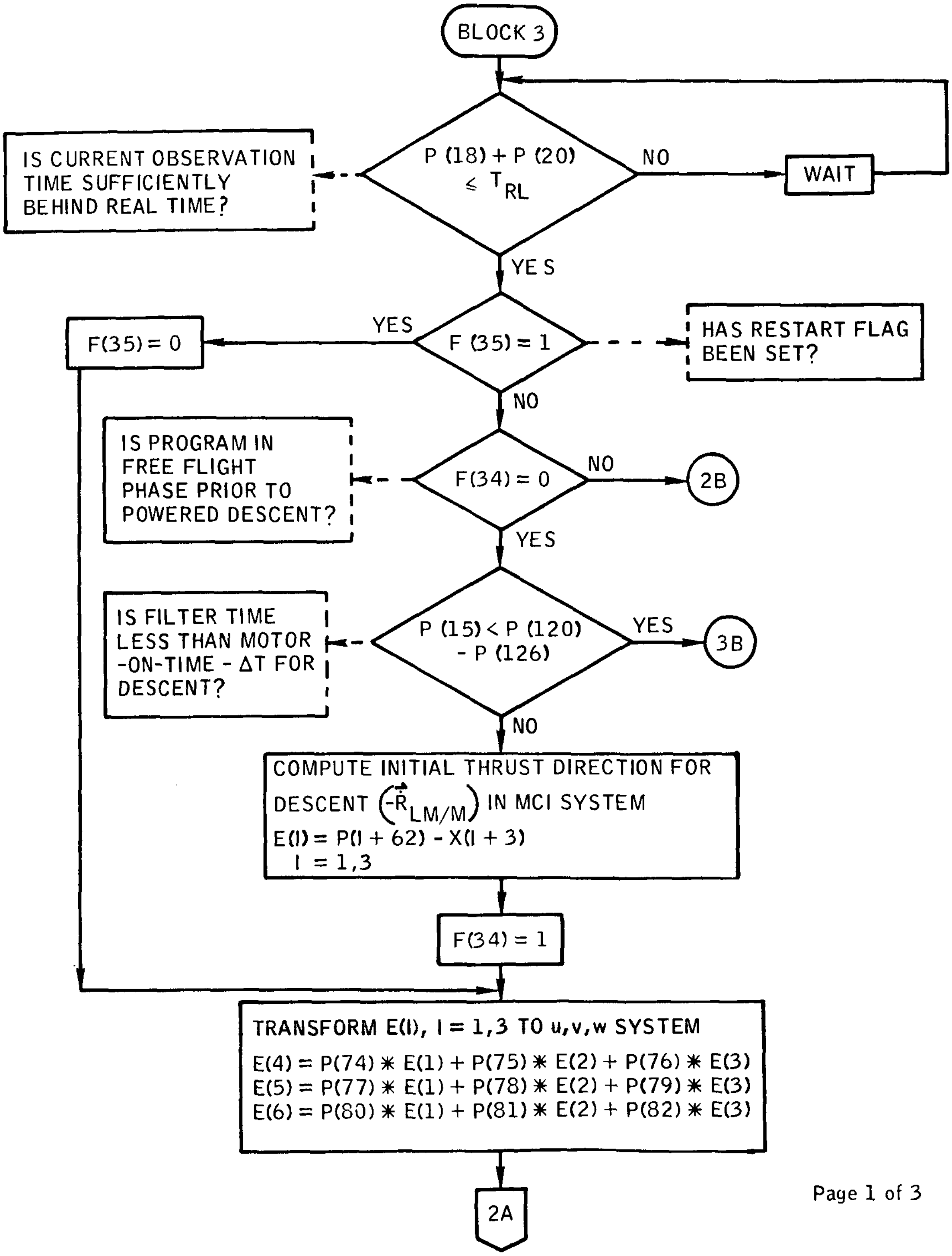


Page 2 of 3

Flow chart 2.- Restart - Continued.

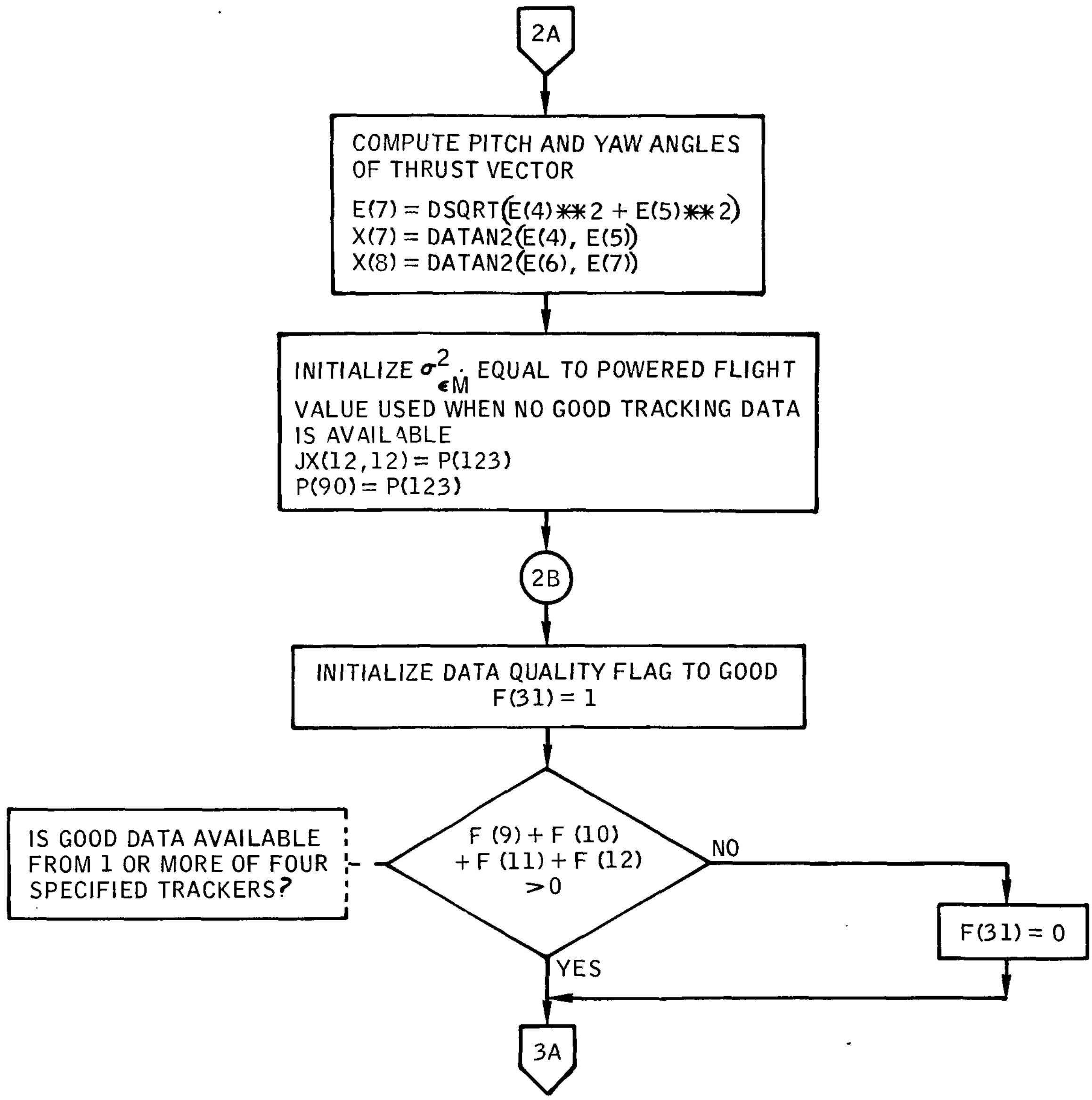


Flow chart 2.- Restart - Concluded.



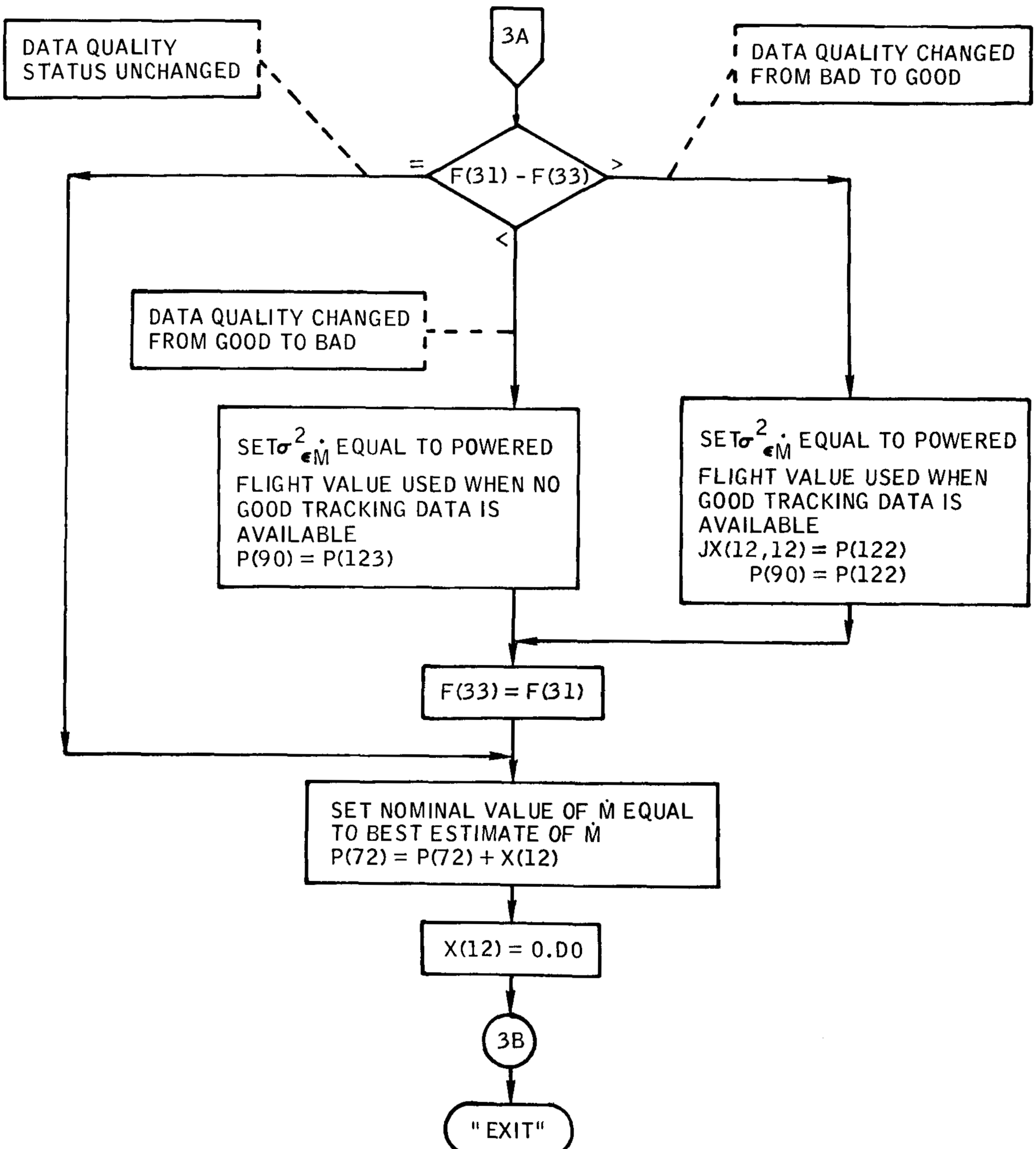
Page 1 of 3

Flow chart 3.- Block 3.



Page 2 of 3

Flow chart 3.- Block 3 - Continued.



Page 3 of 3

Flow chart 3.- Block 3 - Concluded.

REFERENCES

1. Wylie, Alan D.: RTCC Requirements for Mission G: Selecting and Verifying USBS Doppler Data Sources During LM Ascent and Descent. MSC IN 68-FM-105, April 30, 1968.

2. Real-Time Computer Program Requirements for Apollo C-V Volume II, Section TD 2. Philco Report PHO-TR170A.